SIGNAL PROCESSING

FOR

PRECISION WIDESWATH OCEAN BOTTOM

BATHYMETRY

Ravi Sankar

A Thesis

in

The Department

of

Electrical Engineering

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Engineering at
Concordia University
Montreal, Quebec, Canada

April, 1980

ABSTRACT

SIGNAL PROCESSING

FOR

PRECISION WIDE SWATH OCEAN BOTTOM BATHYMETRY

Ravi Sankar

In this thesis, an ocean bottom return signal model for a wide swath multi-beam bottom mapping system is formulated and digital signal processing techniques for mapping applications have been investigated.

First, a mathematical model for representing the smooth envelope function of the bottom return signal received from a number of spatial directions comprising a wide swath is presented. The computer simulation results show that the -3 dB width of the envelope function is closely matched to the expected echo duration for each spatial direction. It is found that with the introduction of background noise and amplitude fluctuations in the expected envelope, the model represents a backscattered signal that would be seen by a bottom mapping system.

Following this, a digital nonrecursive matched filter for processing the ocean bottom return signals has been investigated. Different windows, in both the frequency and time domains, are considered. Based on the results of the investigation it is found that the Tukey window is the best window for mapping applications. For the Tukey window, the tapering coefficient is optimized to obtain the best performance.

The matched filter implementation using high speed convolution is presented. The high speed convolution is synthesized using FFT. Sectioning techniques are used to obtain an efficient processor. The signal processor output results demonstrate the effectiveness of the filter. It is seen from the computer simulation that the background noise is smoothed, while producing a signal return with a well defined peak. The error in [redacted] of arrival is found to be less than 3 ms and the error in slant range is less than 0.1% for an average signal-to-noise ratio of 15 dB and a vertical ocean depth of 12000 ft. These results apply to mapping at angles up to $\pm$ 45° off vertical.

The high speed convolution using special transform techniques has also been studied and compared with the FFT implementation. With special purpose hardware, the multidimensional techniques using the Fermat Number Transform (FNT) provide better computational efficiency than the FFT implementation. It is felt that further improvements in the computational efficiency can be achieved using special transform techniques.

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to his thesis supervisor, Dr. S.D. Morgera, of the Electrical Engineering Department, Concordia University, for initiating the project and providing continued guidance throughout this investigation.

Thanks are also extended to Mr. Harold March for helping to prepare the computer plots and to Ms. Elizabeth Horwood for typing the manuscript.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A_j	number of additions
BW	bandwidth
c	average velocity of sound in water, ft/s
d	operating depth, ft
DFT	discrete Fourier transform
f	frequency, Hz
f_s	sampling frequency, Hz
FFT	Fast Fourier transform
FIR	finite impulse response
FNT	Fermat number transform
h	mean peak - trough roughness
IDFT	inverse discrete Fourier transform
$I_o(\cdot)$	modified Bessel function of the first kind and zero order
J	Jacobian transformation
k	time integer
LPF	lowpass filter
m	frequency integer
M_j	number of multiplications
$p(\cdot)$	probability density function
PDF	probability density function
R	Rayleigh parameter
R_s	slant range, ft

S	input signal-to-noise ratio
SAR	synthetic aperture radar
SNR	signal-to-noise ratio
t	time, s
T_s	sampling period, s
T_w	-40 dB time width, s
$\Phi(\phi,d)$	expected echo duration, s
W_1	bandwidth of lowpass filter (additive noise)
W_2	bandwidth of lowpass filter (multiplicative noise)
α	a constant in the envelope function
α_t	tapering coefficient (Tukey window)
ε	time of arrival error in the matched filter output
θ_R	-3 dB beam width, rad
λ	acoustic wavelength
σ	standard deviation
Σ	indicates summation
τ	time duration of the sinusoidal transmitted pulse, s
Φ	beam steering angle, deg.

The following notations are carried out throughout this thesis:

$h(t)$	any function of the continuous time variable t
$H(f)$	any function of the continuous frequency variable f
$h(k)$	any function of the discrete time variable k
$H(m)$	the discrete Fourier transform of $h(k)$

CHAPTER I.

CHAPTER I

INTRODUCTION

1.1 BACKGROUND

The mapping of the ocean floor topography is gaining increasing importance because of the ever-increasing demands of energy and natural wealth which are abundant, but not yet fully exploited in and under the oceans. A wide swath ocean bottom mapping system is one such method to efficiently provide bathymetric data over large areas of the ocean bottom by obtaining precision maps. The applications of precision ocean mapping are many and varied as outlined in the following points:

i) Topography and Mapping - for development of raw materials from the ocean, installation of equipment and pipelines, and land reclamation.

ii) Positioning and Navigation - both surface and submerged.

iii) Boundary Demarcation and Determination.

iv) Hydrographic and Geological Surveying.

v) Ground Discrimination - for identifying the nature of the bottom.

vi) Recovery and search of underwater objects and equipment.

vii) Ecology - related to garbage disposal and other ocean dumping.

viii) Improved information about the ocean floor and acting as a warning system.

ix) Gravity measurements at ocean floor - for improved datum relationship.

Drummond [1] investigated the current status and requirements for ocean bottom mapping and has pointed out that bathymetric data taken

over some areas of the ocean are old or inadequate and need to be updated with more precision.

1.2 GENERAL DESCRIPTION OF A WIDE SWATH BOTTOM MAPPING SYSTEM

A wide swath bottom mapping system provides bathymetric data over large areas of the ocean bottom. The general structure of such a system is as shown in Figure 1.1. It consists of a data processing system (DPS) which provides overall system control and decision making, a control / display unit, a signal generator, an array for transmitting and receiving acoustic energy, and a signal processor for detecting and estimating the values of the parameters (the time-of-arrival data, depth, steering angle, and the expected echo duration) representing the signal.

The transmitting array of transducers provides a high-intensity acoustic signal with a stabilized fan shaped beam pattern. A parametric array [2] could also be used in this application. The athwartships beam-width is large enough to cover the desired angular swath centered on the vertical underneath the survey vessel. Figure 1.2 shows the swath geometry for the ocean bottom bathymetry. The receive hydrophone array receives the acoustic signals reflected from the ocean bottom swath insonified by the transmitted beam and spatially separates the signals into a number of stabilized preformed beams. The ocean bottom return signal from each beam is then processed by the signal processor to improve the signal-to-noise ratio to extract time-of-arrival data for each beam output. The data processing system could be a general-purpose computer for coordinating the activities of the entire wide swath bottom mapping system. It also receives the processed data from the signal processor and performs

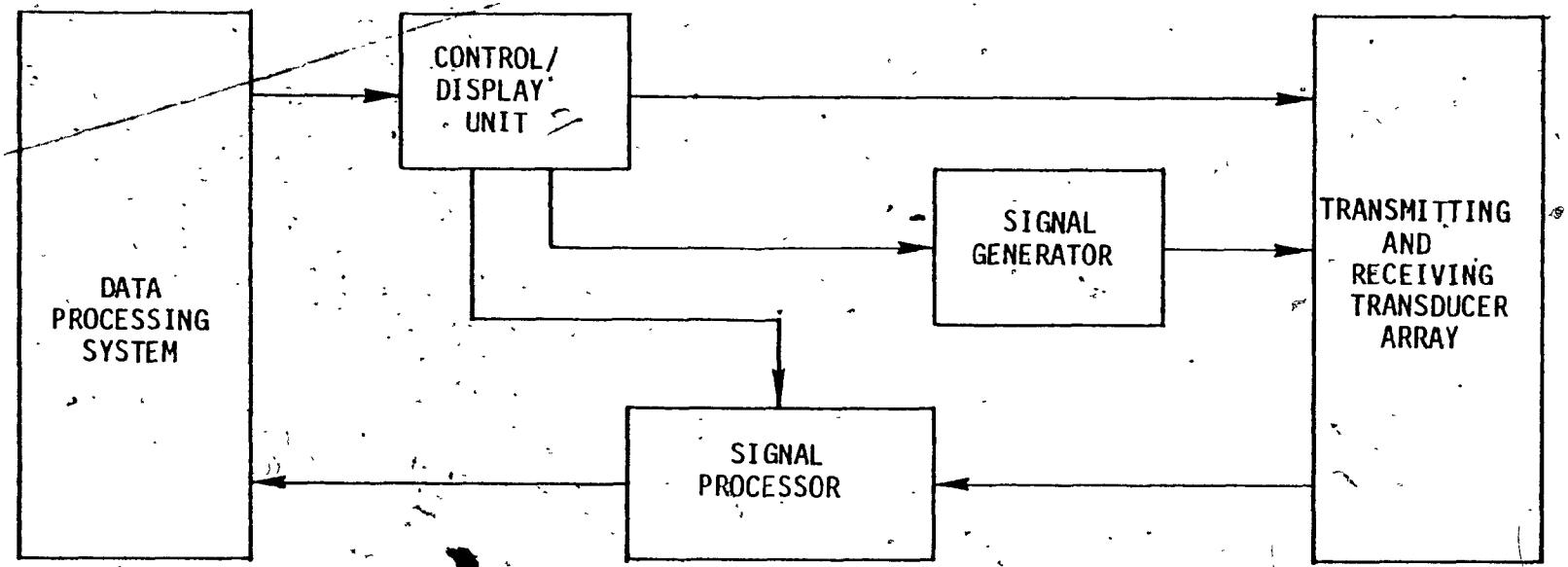


FIGURE 1.1: General Structure of a Battom Mapping System

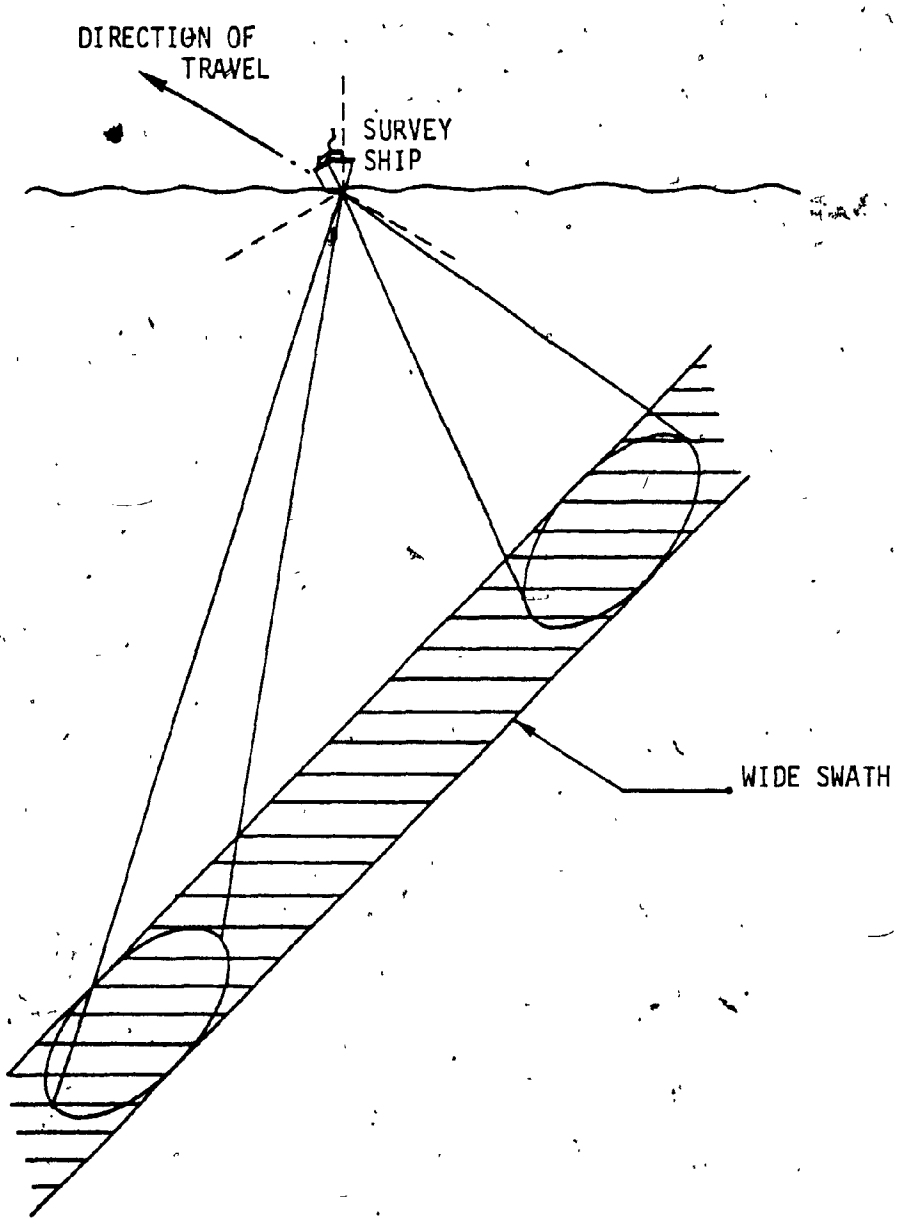


FIGURE 1.2: SWATH GEOMETRY

some special functions like detecting the signal and providing details for display.

The wide swath bottom mapping system is a multi-beam sonar system which provides maps for a wide swath as shown in Figure 1.2. Compared to conventional single beam echo sounding methods providing topographic data only along a single traverse directly beneath the survey ship's track, the techniques presented here are far more efficient.

1.3 REVIEW OF PREVIOUS WORK

The early developments in the field of topography and ocean bottom mapping can be traced out as early as World War I [3]. The fathometer, developed by Harvey Hayes, was one of the earliest instruments used for mapping the contours of the ocean bottom. Then the echo sounder was developed and widely used. Essentially, it has a projector to transmit an acoustic signal, a hydrophone for receiving the reflected signal from the ocean floor and a recorder/display which reads the ocean depth by measuring the elapsed time between the transmission and the return of the echo from the ocean bottom.

The narrow beam echo sounding system came into existence, and used a low-frequency, directional beam to achieve a better resolution. To obtain these characteristics, the transducer has to be large. This problem was overcome by parametric sonar. Westervelt [4] developed the theory of non-linear interaction of two parallel-plane sound waves, which later became the basis of parametric sonar. Parametric sonar has several attractive properties: for example, no sidelobe radiation at the difference frequency, and a narrow beam width. The techniques described

above are essentially single beam mapping systems.

Glenn [5] investigated a multi-narrow beam sonar system. He pointed out that guess work made by the single beam systems in describing the morphology of the feature was eliminated by the use of multi-beam systems. He discussed two commercially available systems, BO'SUN and SEA-BEAM, and he concluded that these systems required advancement in navigation and data processing.

Recently, there has been interest in developing an efficient signal processing for mapping applications. The signal processing is done by two methods, digital or optical. Hogan [6] analysed a Digital Survey Fathometer providing chart display and digital output. It features automatic phasing, time-varying gain circuitry and bottom acquisition gating and provides a high resolution, quality bathymetric profiles. But its operation is limited to 2000 ft in depth.

Brundage and Patterson [7] used optical tools for ocean bottom mapping. They used in-water Light Behind Camera (LIBEC) technique. The mapping of the ocean floor was done with strips of overlapping LIBEC photographs. Such techniques, however, are depth-limited and only work well in clear fluid.

Lee [8] presented a synthetic aperture radar (SAR) technique to underwater mapping applications. He pointed out that map resolutions, both in azimuth and in range were derived using the SAR technique. Various methods were discussed in his paper to significantly improve the system mapping rate. The synthetic array processing which is basically a matched-filter operation, followed by lowpass filtering.

The output of the filter is linearly detected, logarithmically compressed, and outputted to the display console. He considered a digital means of processing rather than an optical method because of its flexibility, small size, and adaptability to real-time operation.

Folds and Anderson [9] developed a mathematical model for a computer simulation of both the side-looking and forward-looking high resolution sonar systems.

Bartram [10] presented a method of estimating the Doppler frequency shift of an echo reflected from the ocean bottom. He made use of the possible alternative signal characteristics of the signal spectrum by representing the Doppler spectrum by a Gaussian shape. He derived a signal processor by maximum likelihood methods and evaluated its performance under ideal conditions. This work is considered a valuable and timely contribution to the modeling of ocean bottom scattered signals.

Morgera [11] presented a signal processing technique for multibeam precise ocean mapping. The bottom return signal model and accompanying signal processor for a wide swath bottom mapping system were described. He modeled the expected envelope of the signal received from various spatial directions as a smooth nearly Gaussian-shaped function for near-vertical steering angles and a nearly Rayleigh-shaped function for off-vertical angles. The signal processor was essentially a matched filter matched to the received signal envelope for each of the steering angles. He derived the matched filter by using stages of cascaded one-pole recursive filters to obtain an impulse response of the proper shape for each steering angle [12]. The technique is one which appears to

provide accurate wide swath maps with a minimum of processing complexity. In addition to not having the vehicle speed-dependent accuracy of the SAR technique, the amount of hardware required for implementation is considerably reduced.

1.4 SCOPE OF THE RESEARCH WORK

The scope of this thesis is to formulate an ocean bottom return signal model and develop an efficient digital signal processing technique for a wide swath bottom mapping system.

A mathematical model representing the expected envelope of the signal received from the various spatial directions is derived as a smooth bottom function representing the effects of the two-way (transmit-receive) spatial beam pattern, angle of incidence, and depth in Chapter II. A characterization of the bottom return signal is obtained by corrupting the smooth envelope function with additive and multiplicative noise disturbances. A computer simulation of the bottom return signal model is also outlined in Chapter II.

In Chapter III, a brief study of matched-filter theory is presented. Windowing is performed to obtain a finite impulse response of the matched filter. Different window functions both in the frequency and time domain are considered, and the optimal window for the mapping application is selected. The techniques presented here differ from those of [11] in that we employ a non-recursive matched filter for the various spatial angles, rather than a recursive matched filter.

The matched-filter implementation using High Speed Convolution is considered in Chapter IV. First the high-speed implementation using

fast Fourier transform (FFT) is presented. Sectioning techniques are used with FFT to derive an efficient method of processing. The high speed convolution using some special transforms is considered and compared with the FFT implementation.

In Chapter V, the performance evaluation of the overall system is provided. The simulation results describing the model and the signal processor are provided with descriptive figures and tables.

Finally, in Chapter VI a conclusion on signal processing for precision wide swath ocean bottom bathymetry is drawn from the simulation and the signal processing techniques adopted. Recommendations for further research are outlined.

CHAPTER II

CHAPTER II

MATHEMATICAL MODELING AND SIMULATION OF A BOTTOM RETURN SIGNAL

2.1 BACKGROUND

In this chapter, a realistic mathematical model is developed for the ocean bottom return signal. The Rician probability density function (PDF) commonly used in communications to describe a slow fading channel [13,14] is made use of as a starting point in the formulation of the model. By selecting new sets of parameters in the Rician PDF, the functional form is seen to closely model the envelope of a signal backscattered into various spatial directions from the ocean bottom. Simulation is carried out using a digital computer to investigate the performance of the model.

The wide swath bottom mapping system is assumed to be equipped with a high resolution beam former [15]. The output of all the beams are modeled as a signal backscattered from the random rough sea floor except those oriented within the vicinity of normal incidence to the bottom where the rough surface becomes effectively smooth. The criterion for the roughness of the sea floor is given by the Rayleigh parameter, defined as $R = 2h\sin\theta/\lambda$, where θ is the angle of incidence, λ is the acoustic wavelength and h is the mean peak-trough roughness. When $R \gg 1$, the sea floor acts as a scatterer, sending incoherent energy in all directions representing a random rough floor [2]. The spatial correlation distance of the bottom roughness is assumed to be negligible relative to the insonified area. In this work a high intensity sinusoidal burst, relatively short in duration is used as transmit signal. The envelope may be rectangular, as assumed here, or shaped, if this is desirable.

The above assumptions permit representing the backscattered energy as a burst of narrow band noise, amplitude weighted as a function of time by the intersection of the spatial beam pattern with the bottom. The envelope of the backscattered signal is best modeled as a deterministic function, the shape of which depends on the steering angle (Φ) and depth (d) and having both additive and multiplicative noise disturbances. If the spatial beam pattern is approximated by a Gaussian function, the envelope of the bottom return signal for the beam near the vertical appears to have a Gaussian shape with a smooth transition to an approximately Rayleigh-shape for larger steering angles as shown in Figure 2.1. The peak of the bottom return signal envelope corresponds to the maximum response axis (MRA) of the beam. We note that the envelope function and its dependence on steering angle and depth closely resemble the Rician probability density function (PDF) in shape [1].

2.2 MATHEMATICAL MODELING OF THE BOTTOM RETURN SIGNAL

A mathematical model for the bottom return signal for a wide swath bottom mapping system is developed. The envelope of the signal received from the various spatial directions is modeled as a smooth, nearly Rician function representing the effects of the two-way spatial beam pattern, angle of incidence, and depth.

2.2.1 Probability Density Function for the Envelope of Sine Wave Plus Noise [6-18]

Consider a carrier wave $E_c \cos\omega_c t$ with a white Gaussian noise passed through a narrow band filter followed by a linear envelope detector as shown in Figure 2.2. The output signal of the narrow band filter without the noise can be written as:

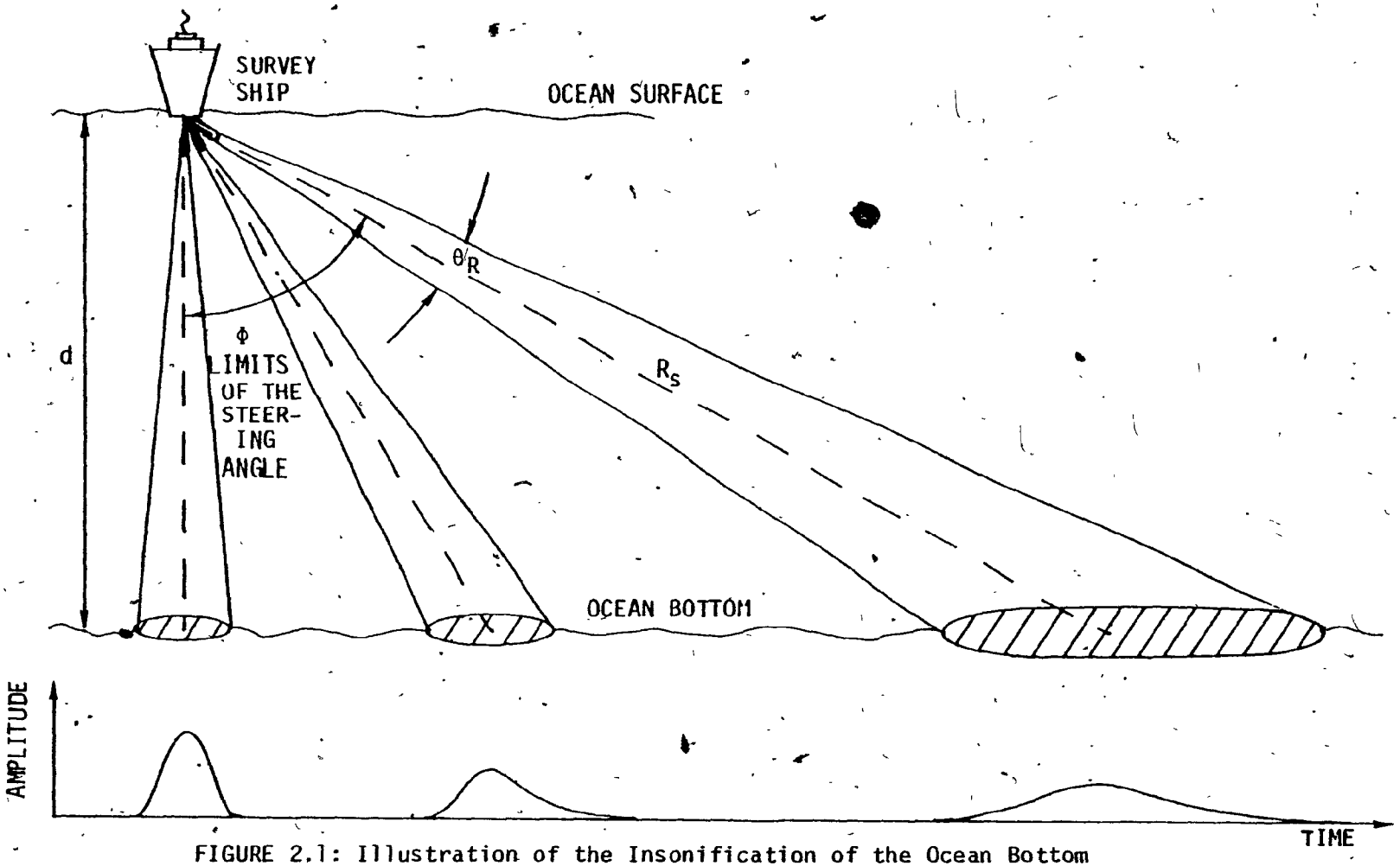


FIGURE 2.1: Illustration of the Insonification of the Ocean Bottom and the Corresponding Envelope Function

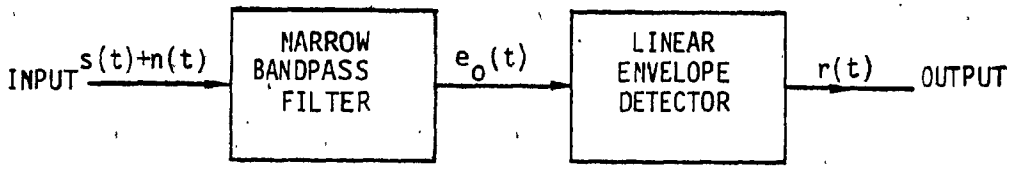


FIGURE 2.2: Narrow Bandpass Filter followed by Linear Envelope Detector

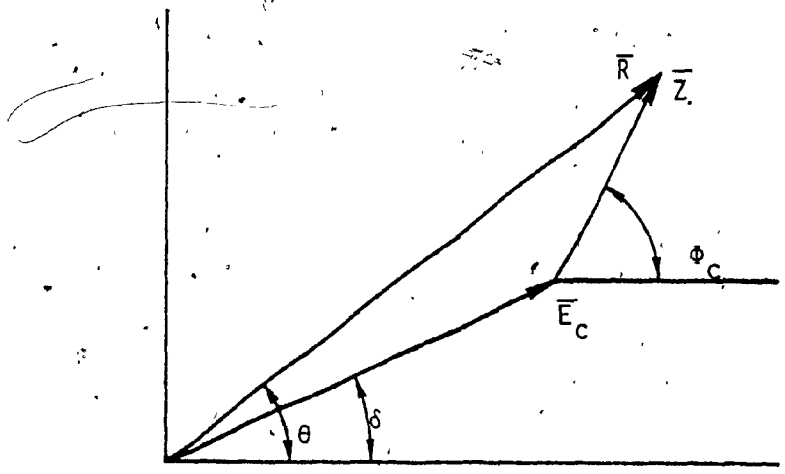


FIGURE 2.3: Vector Representation for the Envelope of Sine Wave Plus Noise

$$S_0(t) = E_c \cos(\omega_c t + \delta) \tag{2.1}$$

Rewriting equation (2.1)

$$S_0(t) = E_{1c} \cos \omega_c t + E_{2c} \sin \omega_c t \tag{2.2}$$

where $E_c = [E_{1c}^2 + E_{2c}^2]^{\frac{1}{2}}$

and $\delta = \tan^{-1}[E_{2c}/E_{1c}]$

The output noise of the narrow band filter will be Gaussian and can be represented as

$$n_0(t) = X(t) \cos \omega_c t + Y(t) \sin \omega_c t \tag{2.3}$$

$$= Z(t) \cos(\omega_c t + \Phi_c)$$

where $X(t)$ and $Y(t)$ are independent normally distributed (Gaussian) random variables with zero mean and variances σ^2, respectively

$$Z(t) = [X^2(t) + Y^2(t)]^{\frac{1}{2}}$$

and $\Phi_c = \tan^{-1}[Y(t)/X(t)]$

Now the signal plus noise at the filter output $e_0(t)$ can be obtained by adding equations (2.2) and (2.3):

$$e_0(t) = r(t) \cos(\omega_c t + \theta) \tag{2.4}$$

where $[r(t)]^2 = [E_{1c} + X(t)]^2 + [E_{2c} + Y(t)]^2$

and $\theta = \tan^{-1}\left[\dfrac{E_{2c} + Y(t)}{E_{1c} + X(t)}\right]$

Equation (2.4) represents the envelope-phase of the output signal of the narrow band filter. If $e_0(t)$ is applied to a linear envelope detector, the output will be $r(t)$, the envelope of the input to the detector.

From the probability density function (PDF) of $X(t)$ and $Y(t)$, the PDF of the output $r(t)$ can be evaluated.

$$\text{Let } r(t) = [x_1^2(t) + y_1^2(t)]^{\frac{1}{2}} \tag{2.5}$$

where $x_1(t) = E_{1c} + X(t)$

and $y_1(t) = E_{2c} + Y(t)$

Since $X(t)$ and $Y(t)$ are independent normally distributed random variables, $x_1(t)$ and $y_1(t)$ are also independent normally distributed random variables with mean values E_{1c} and E_{2c}, and variances σ^2 respectively.

The relationship among the quantities (r, θ),(E_c, δ) and (Z, Φ_c) are vectorially shown in Figure 2.3. Thus the probability densities of the random variables x_1 and y_1 are

$$p(x_1) = \frac{1}{[2\pi\sigma^2]^{\frac{1}{2}}} \exp\left[-\frac{(x_1 - E_{1c})^2}{2\sigma^2}\right] \tag{2.6}$$

$$p(y_1) = \frac{1}{[2\pi\sigma^2]^{\frac{1}{2}}} \exp\left[-\frac{(y_1 - E_{2c})^2}{2\sigma^2}\right] \tag{2.7}$$

Then the joint probability density function $p(x_1, y_1)$, is the product of the two first order Gaussian probability densities $p(x_1)$ and $p(y_1)$:

$$p(x_1, y_1) = p(x_1) \cdot p(y_1) \tag{2.8}$$

$$= \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x_1 - E_{1c})^2 + (y_1 - E_{2c})^2}{2\sigma^2}\right]$$

By using $x_1 = r \cos\theta$ and $y_1 = r \sin\theta$

the joint PDF can be expressed as

$$p(x_1, y_1)\, dx_1\, dy_1 = p(r, \theta)\, dr d\theta \tag{2.9}$$

The area elements in the two co-ordinate systems can be expressed in terms of the Jacobian as:

$$dr d\theta = J\,[(r,\theta)/(x_1,y_1)]\; dx_1 dy_1$$

where J is the Jacobian of the transformation given by the determinant

$$J\,[(r,\theta)/(x_1,y_1)] = \begin{vmatrix} \frac{\partial r}{\partial x_1} & \frac{\partial r}{\partial y_1} \\ \frac{\partial \theta}{\partial x_1} & \frac{\partial \theta}{\partial y_1} \end{vmatrix} = \frac{1}{r}$$

Thus transforming the differential area $dx_1 dy_1$ to $r\, dr d\theta$, the joint PDF is

$$p(r,\theta) = [(r)/(2\pi\sigma^2)] \exp\left[-\frac{(r^2+E_{1c}{}^2+E_{2c}{}^2-2rE_{1c}\cos\theta-2rE_{2c}\sin\theta)}{2\sigma^2}\right] \tag{2.10}$$

Letting $E_{1c}\cos\theta+E_{2c}\sin\theta = E_c\cos(\theta-\delta)$,

$$p(r,\theta) = \begin{cases} \frac{r}{2\pi\sigma^2} \exp\left[-\frac{r^2+E_c{}^2-2rE_c\cos(\theta-\delta)}{2\sigma^2}\right], & \left\{\begin{matrix} 0 \le a \le \infty \\ 0 \le \theta-\delta \le 2\pi \end{matrix}\right\} \\ 0, & \text{elsewhere} \end{cases} \tag{2.11}$$

where $r = (x_1{}^2+y_1{}^2)^{\frac{1}{2}}$ and $\theta = \tan^{-1}(y_1/x_1)$

Unlike x_1 and y_1, r and θ are not statistically independent because of the term $2rE_c\cos(\theta-\delta)$ in the exponent of equation (2.11). By integrating equation (2.11) over θ, $p(r)$ can be obtained as:

$$p(r) = \int_0^{2\pi} p(r,\theta)\, d\theta$$

$$= \int_0^{2\pi} \frac{r}{2\pi\sigma^2} \exp\left[-\frac{r^2+E_c{}^2-2rE_c\cos(\theta-\delta)}{2\sigma^2}\right] d\theta \tag{2.12}$$

Equation (2.12) gives the PDF for the envelope r(t):

$$p(r) = (r/\sigma^2)\exp\left[-(r^2+E_c{}^2)/(2\sigma^2)\right] I_o[(rE_c)/(\sigma^2)] \tag{2.13}$$

where $I_o(x)$ is the modified Bessel function of the first kind and zero order and defined as the definite integral -

$$I_o(x) = 1/2\pi \int_0^{2\pi} e^{x\cos\theta}\, d\theta = \sum_{n=0}^{\infty} (x^{2n}/2^{2n})(n!)^2$$

Equation (2.13) is known as the Rician probability density function.

2.2.2 Limiting Forms for the Rician PDF

Let S be the signal to noise ratio (SNR) at the input to the detector. Then

$$S = E_c^2/2\sigma^2 \tag{2.14}$$

Now in terms of the input SNR, the PDF at the output of the linear envelope detector is given by

$$p(r) = (r/\sigma^2)\exp\left[-(r^2/2\sigma^2) - S\right] I_o\left[(r/\sigma)(2S)^{\frac{1}{2}}\right] \tag{2.15}$$

For S = 0,

$$p(r) = (r/\sigma^2)\exp(-r^2/2\sigma^2) \tag{2.16}$$

Using the approximations of the Bessel function $I_o(x)$:

$$I_o(x) \simeq \begin{cases} e^{x^2/4} & , x<<1 \\ e^x/(2\pi x)^{\frac{1}{2}} & , x>>1 \end{cases} \tag{2.17}$$

Therefore, equation (2.15) approximates to

$$p(r) \simeq \begin{cases} (r/\sigma^2)\exp(-r^2/2\sigma^2) & , s<<1 \quad (2.18) \\ [1/(2\pi\sigma^2)^{\frac{1}{2}}]\exp[-(r-E_c)^2/2\sigma^2] & , s>>1 \quad (2.19) \end{cases}$$

The equations (2.18) and (2.19) can be observed to be Rayleigh and Gaussian density functions, respectively. Thus it can be seen that the output signal density function (Rician PDF) tends to a Rayleigh function for $s<<1$ (i.e., presence of narrow-band Gaussian noise only) and becomes a Gaussian function for $s>>1$ (i.e. presence of signal with increased power) and is illustrated in Figure 2.4.

2.3 SYNTHESIS OF BOTTOM RETURN SIGNAL USING THE RICIAN FUNCTIONAL FORM

The family of curves representing the Rician function can approximate the envelope of the signal received from the ocean bottom by change of parameters in the Rician PDF (2.15). In the limits $s>>1$ and $s<<1$, the Rician PDF represents the envelope of the bottom return signal for near vertical steering angle ($\Phi \approx 0$) and for larger steering angles, respectively.

Let E_c and σ be assumed to be a function of beam steering angle relative to the vertical (Φ) and depth of the ocean (d) (cf. Figure 2.1) as:

$$E_c = f(\Phi)$$

and $$\sigma = g(\Phi, d)$$

Now referring to equations (2.16)-(2.19), it can be seen that for increasing values of steering angle Φ, E_c must decrease and σ must increase in order to achieve the desired model.

Therefore, for $\Phi = 90$ deg.

$$S = E_c^2/2\sigma^2 = 0 \text{ or } E_c = 0 \qquad (2.20)$$

and for $\Phi = 0$ deg,

$$S = E_c^2/2\sigma^2 >> 1 \qquad (2.21)$$

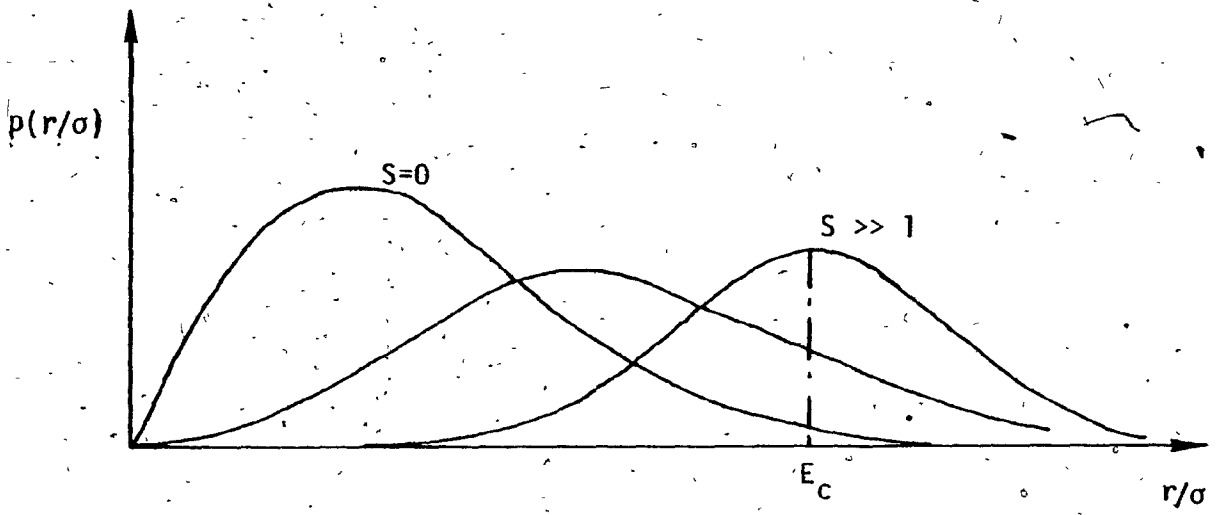


FIGURE 2.4: Rician Probability Density Function for the Envelope of Sine Wave Plus Noise

It is possible to select a function $f(\Phi)$ of the form:

$$f(\Phi) = \tau[\exp(\cos\Phi) - 1] \tag{2.22}$$

which would satisfy the conditions in equations (2.20) and (2.21). Equation (2.22) is one of many possible functions that will satisfy such a functional relationship. It should be noted that in equation (2.22), τ corresponds to the transmitted signal time duration.

The dependence of the standard deviation σ on Φ and d can be expressed [11,19]:

$$\sigma = \alpha\, T(\Phi,d) \tag{2.23}$$

where α is a constant chosen such that $T(\Phi,d)$, the expected echo duration, is nearly equal to the -3dB width of the envelope of the bottom return signal [11].

The expression for $T(\Phi,d)$ can be obtained from the geometric construction shown in Figure 2.5 [21]. Consider a beam pointing at an angle Φ degree from the vertical and θ_R is the beam width angle. The center of the beam strikes at D, the inner edge at B and outer edge at C. BE represents the wave front. The points B and E on the incoming wavefront which traveled together at launch time arrive back at the receiver separated from each other by the propagation time to traverse distance 2EC. The received echo pulse has therefore been stretched. The time stretching does not occur when EC becomes small or very near to zero. That is for vertical steering angle ($\Phi \approx 0$), EC vanishes, thereby the received echo duration is almost equal to the time duration of the transmitted pulse. Therefore, time duration due to stretching = 2EC/c, c is the average velocity of sound in water.

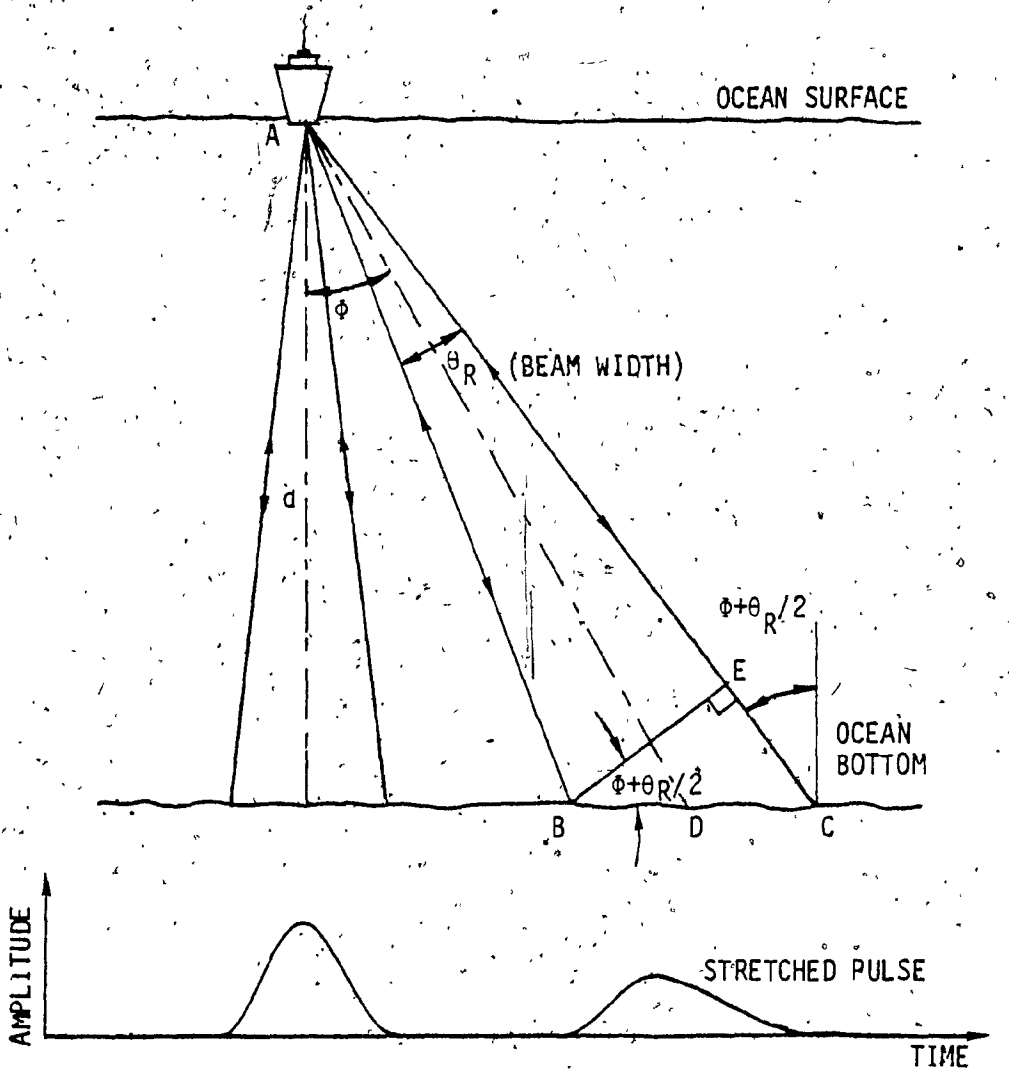


FIGURE 2.5: Geometrical Construction for Derivation of Expected Echo Duration, $T(\Phi,d)$

Using geometrical relationship in Figure 2.5

$$\frac{2\,EC}{c} = \frac{2\,BC}{c} \cdot \sin(\Phi + \theta_R/2)$$

$$= \frac{2}{c} \cdot \frac{AC \sin\theta_R}{\cos(\Phi - \theta_R/2)} \cdot \sin(\Phi + \theta_R/2)$$

$$= \frac{2}{c} \cdot \frac{d}{\cos(\Phi + \theta_R/2)} \cdot \frac{\sin\theta_R}{\cos(\Phi - \theta_R/2)} \cdot \sin(\Phi + \theta_R/2)$$

$$= \frac{2d}{\cos(\Phi - \theta_R/2)} \cdot \frac{\sin\theta_R}{c} \cdot \tan(\Phi + \theta_R/2)$$

Since θ_R is very small ($\theta_R \simeq 1°$) and $\theta_R << \Phi$, the expected echo duration can be approximated to

$T(\Phi,d) \simeq$ time duration due to stretching + time duration of the transmitted pulse

$$\simeq (2d/c)\theta_R \tan\Phi \sec\Phi + \tau \qquad (2.24)$$

where $T(\Phi,d)$ is the expected echo duration , seconds

c is the average velocity of sound in water , ft/s

Φ is the beam steering angle , deg.

d is the operating depth , ft.

θ_R is the -3dB down beam width , rad.

and τ is the time duration of the sinusoidal probe pulse , seconds.

Once the expression for $T(\Phi,d)$ is obtained, the constant α has to be evaluated for all the beam steering angles.

For Near Vertical Steering Angles ($\Phi \simeq 0$):

In the case of near vertical steering angles, it has been shown

that the Rician function approximates a Gaussian PDF. Hence from equation (2.19)

$$p(r)\Big|_{r = E_c + a\sigma} \simeq [1/(2\pi\sigma^2)^{\frac{1}{2}}] \exp(-a^2/2) \qquad (2.25)$$

Now referring to Figure 2.6, $p(r)\Big|_{r = E_c + a\sigma} = 0.707$

Therefore, the value of "a" can be calculated as 0.833.

Since T (Φ,d) is the expected echo duration corresponding to -3db width,

$$T(\Phi,d) \simeq 2a\sigma$$

Also $T(\Phi,d) \simeq \sigma/\alpha$

Therefore, $\alpha \simeq 1/2a \simeq 0.6$ (2.26)

For Steering Angles Away from Vertical

In this case, the Rician function approximates a Rayleigh function as given in equation (2.18). Now referring to Figure 2.7, the values of r_1 and r_2 corresponding to 0.707 of the peak value (-3db down) can be obtained by substituting $r = a\sigma$ in equation (2.18). That is

$$p(r)\Big|_{r=a\sigma} = (a/\sigma) \exp(-a^2/2) \simeq 0.707/\sigma(e)^{\frac{1}{2}} \qquad (2.27)$$

From equation (2.25), there will be two values of "a" corresponding to r_1 and r_2, and they are $r_1 \simeq 0.483$ and $r_2 \simeq 1.637$.

Hence

$$T(\Phi,d) \simeq (r_2 - r_1)\sigma$$
$$\simeq \sigma/\alpha \qquad (2.28)$$

Therefore

$$\alpha \simeq 1/(r_2 - r_1) \simeq 0.8$$

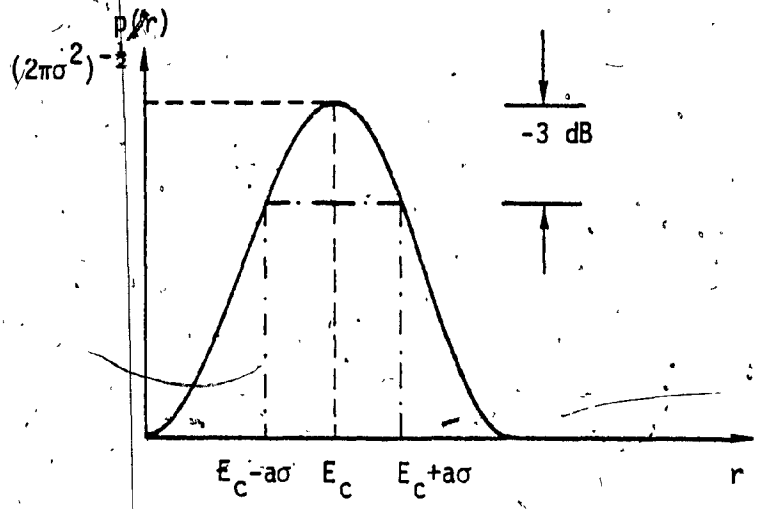


FIGURE 2.6: The Gaussian Function

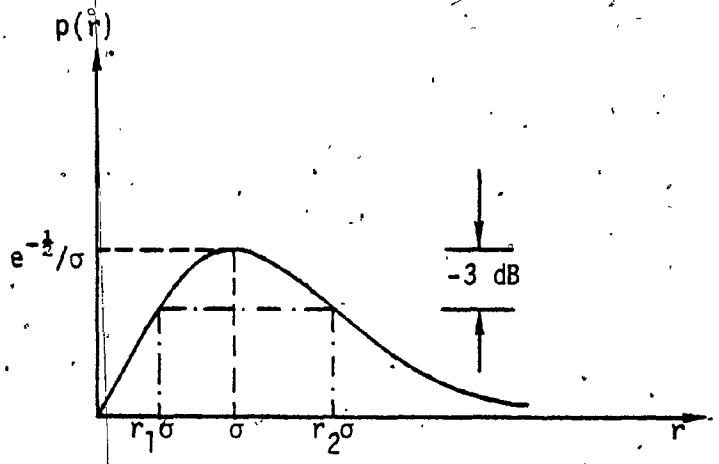


FIGURE 2.7: The Rayleigh Function

2.4 SUMMARY OF THE MATHEMATICAL MODEL

From the preceding sections it is evident that the Rician functional form represents the envelope of the received backscattered signal from the ocean bottom and is given by:

$$h(t) = (t/\sigma^2) \exp\ -[(t^2+E_c^2)/2\sigma^2]\quad I_0(tE_c/\sigma^2) \tag{2.29}$$

where $h(t)$ is the envelope of the bottom return signal

$$E_c = \tau\ [\ \exp(\cos\Phi)-1\ ] \tag{2.30}$$

$$\sigma = \alpha\ [\theta_R \cdot (2d/c) \tan\Phi\ \sec\Phi\ + \tau\ ] \tag{2.31}$$

$$\alpha = \begin{cases} 0.6, \text{ (near vertical)} \\ \\ 0.8, \text{ (otherwise)} \end{cases} \tag{2.32}$$

(cf. Table 21)

In the model, several parameters can be adjusted to obtain a realistic signal, depending on specific geometric and geographic conditions.

2.5 INTRODUCTION OF NOISE

Once the smooth envelope function of the bottom return signal is developed, a more sophisticated and realistic model is obtained by introducing fluctuation noise into the model thereby adequately representing the characteristics of a sonar channel. Under actual operating conditions, our assumptions of the scattering process and an examination of the real data obtained from the ocean bottom, indicate that the envelope is corrupted by both additive and multiplicative noise disturbances [11, 19-20].

Generally, the unwanted noises corrupting the acoustic signal are classified as ambient noise and reverberant noise. Ambient noise is the background noise, generally additive, that is generated by the numerous other sources of acoustic energy in the ocean. These are noises that are everpresent in the ocean whether the signal is transmitted or not. The reverberant noise arises as a consequence of the transmission. This includes the multipath echoes from the random-rough sea floor, sea surface and the reflections from the numerous particles suspended in the water. The net effect of the sea floor scattering, the reverberant noise of principal concern to us here, is to introduce fluctuations in the bottom return smooth envelope which seem best introduced in a multiplicative fashion [2].

Let the smooth bottom function be sampled at a rate sufficiently higher than the Nyquist rate, then the k th time sample-b(k) of the bottom return signal is given by

$$b(k) = g_1(k)h(k) + g_2(k) \tag{2.33}$$

where $h(k)$ is the discrete smooth bottom function.

$g_1(k)$ is the output of the band limiting digital lowpass filter for the multiplicative noise,

$g_2(k)$ is the output of the bandlimiting digital lowpass filter for the additive noise.

The output of the filters are given by

$$g_1(k) = n_1(k) * f_1(k)$$

$$\text{and} \quad g_2(k) = n_2(k) * f_2(k) \tag{2.34}$$

where $n_1(k)$ and $n_2(k)$, $k = 1,2,\ldots K$ are statistically independent

white sequences which are normally distributed with mean zero and variance σ_1^2 and σ_2^2, respectively and * denotes the convolution operation. The discrete functions f_{1k}, f_{2k} are the sampled impulse responses of the digital lowpass filters used to bandlimit the multiplicative and additive noise disturbances.

The filter f_1 is a function of the signal bandwidth and f_2 is a function of the input noise bandwidth. The variance σ_1^2 of the multiplicative noise disturbance n_1 controls the signal level and the bandwidth w_1 of filter f_1 sets the fluctuation rate of the bottom return signal. The variance σ_2^2 of the additive noise disturbance n_2 controls the background noise level and in conjunction with the bandwidth w_2 of the filter f_2 establishes the input SNR [11]. The schematic diagram shown in figure 2.8 outlines the method used in the simulation of the bottom return signal.

2.6 SIMULATION OF THE MODEL

The discrete smooth bottom return signal envelope

$$h(k) = \frac{k}{\sigma^2} \exp\left[-(k^2+E_c^2)/2\sigma^2\right] \; I_0\left(\frac{kE_c}{\sigma^2}\right)$$

where $E_c = \tau \cdot [\exp(\cos\Phi) - 1]$ and $\sigma = \alpha \left[\theta_R \cdot \frac{2d}{c} \cdot \frac{\tan\Phi}{\cos\Phi} + \tau\right]$

is simulated using the Digital Computer (CDC CYBER 172) to verify the model. All the programs and their brief descriptions are given in the Appendix C. Experimental data have been previously obtained, so that the simulated model for the backscattered signal from the bottom surface is realistic and reasonable [11]. The values of the parameters chosen for realizing the smooth bottom function are:

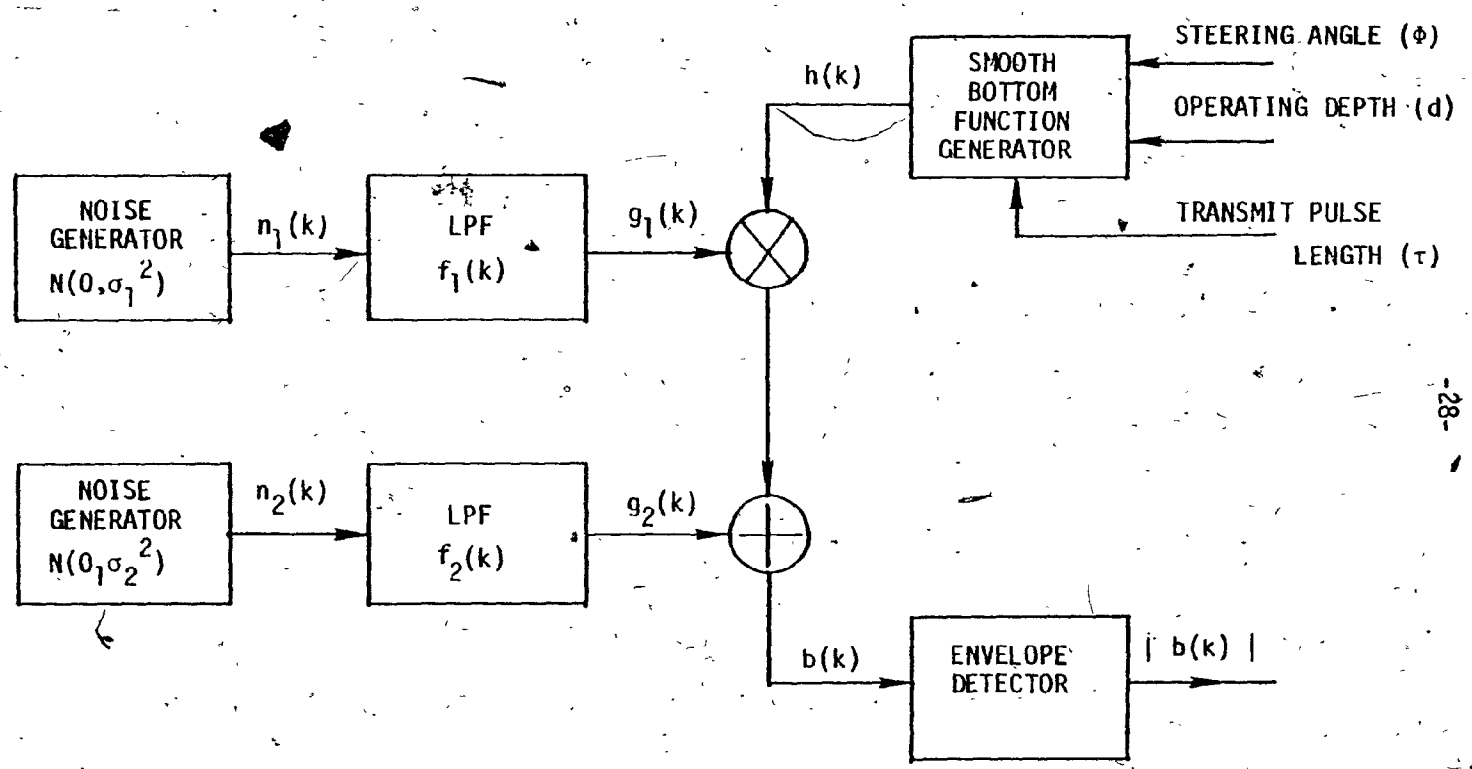


FIGURE 2.8: Simulation of the Bottom Return Signal

The acoustic signal beam width (θ_R) ≈ 1° or $\pi/180$ rad.

Operating depth (d) = 2000 fathoms or 12000 ft.

The pulse length (τ) = 0.01 S.

The average velocity of sound in water (c) = 4800 ft/s.

The steering angle of the beam for 0 degree and 15, 30 and 45 degrees on either side of the vertical are considered. The values of E_c and σ are calculated for all steering angles and then h(k), the discrete samples of the smooth bottom function for the corresponding time integer k[#] are obtained for each steering angle. It is seen already that for steering angles away from vertical, pulse stretching occurs. So if number of samples (K) is kept constant, then, for increasing steering angles the sampling time increases (cf. Table 2.2). The plots of the smooth bottom function for the various steering angles are shown in Figures 2.9-2.13.

In the model we use

$$\sigma = \alpha T(\Phi,d)$$

where $T(\Phi,d)$ is the expected echo duration.

The optimum values of α are chosen to get the best possible results in obtaining the -3dB bandwidth of the envelope very nearly equal to the theoretical values. These values of α are given in Table 2.1.

[#] The time integer k denotes the exact sampling time, kT_S where the sampling time $T_S = 1/f_S$ and f_S is the sampling frequency.

TABLE 2.1

VALUES OF α FOR SELECTED STEERING ANGLES

STEERING ANGLE Φ, DEGREES	α (EXACT)	α (OPTIMUM)
0	0.6	0.65
15	0.8	0.7
30	0.8	0.75
45	0.8	0.8

TABLE 2.2

SMOOTH BOTTOM FUNCTION CHARACTERISTICS FOR VARIOUS STEERING ANGLES

STEERING ANGLE Φ, deg	SAMPLING FREQUENCY f_s, Hz	PEAK AMPLITUDE A_p, Volts	TIME AT PEAK AMPLITUDE t_p, secs.	EXPECTED ECHO DURATION $T(\Phi,d)$, secs.	-3 dB BANDWIDTH secs.	-40 dB BANDWIDTH secs.	NUMBER OF SAMPLES K
0	6960	63.5367	0.0184	0.01	0.0105	0.0368	256
15	2750	22.808	0.0276	0.032	0.031	0.093	256
30	1380	11.65	0.053	0.063	0.0602	0.1855	256
45	670	5.669	0.110	0.123	0.123	0.382	256

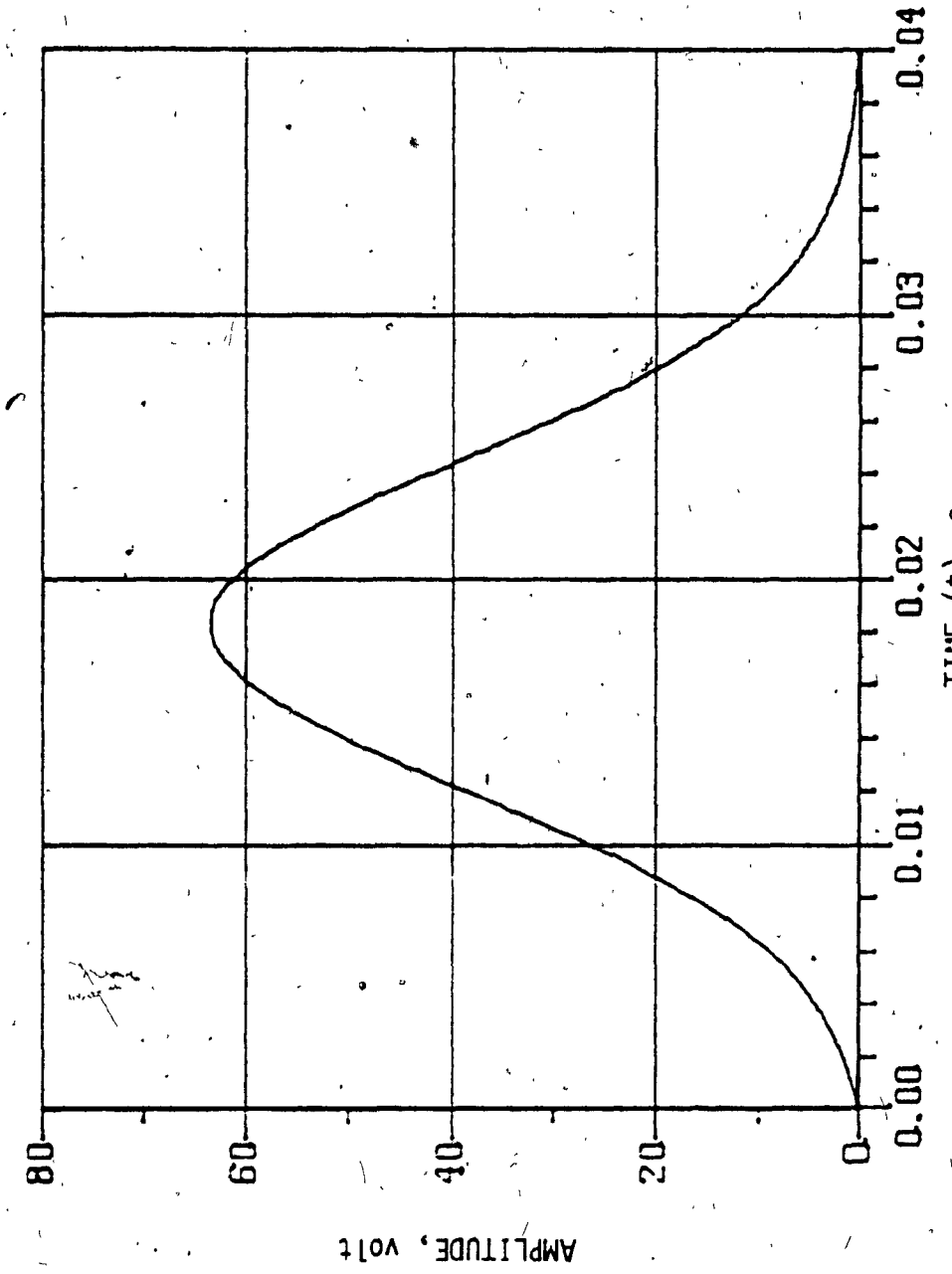


FIGURE 2.9: SMOOTH BOTTOM FUNCTION FOR Φ = 0 deg.

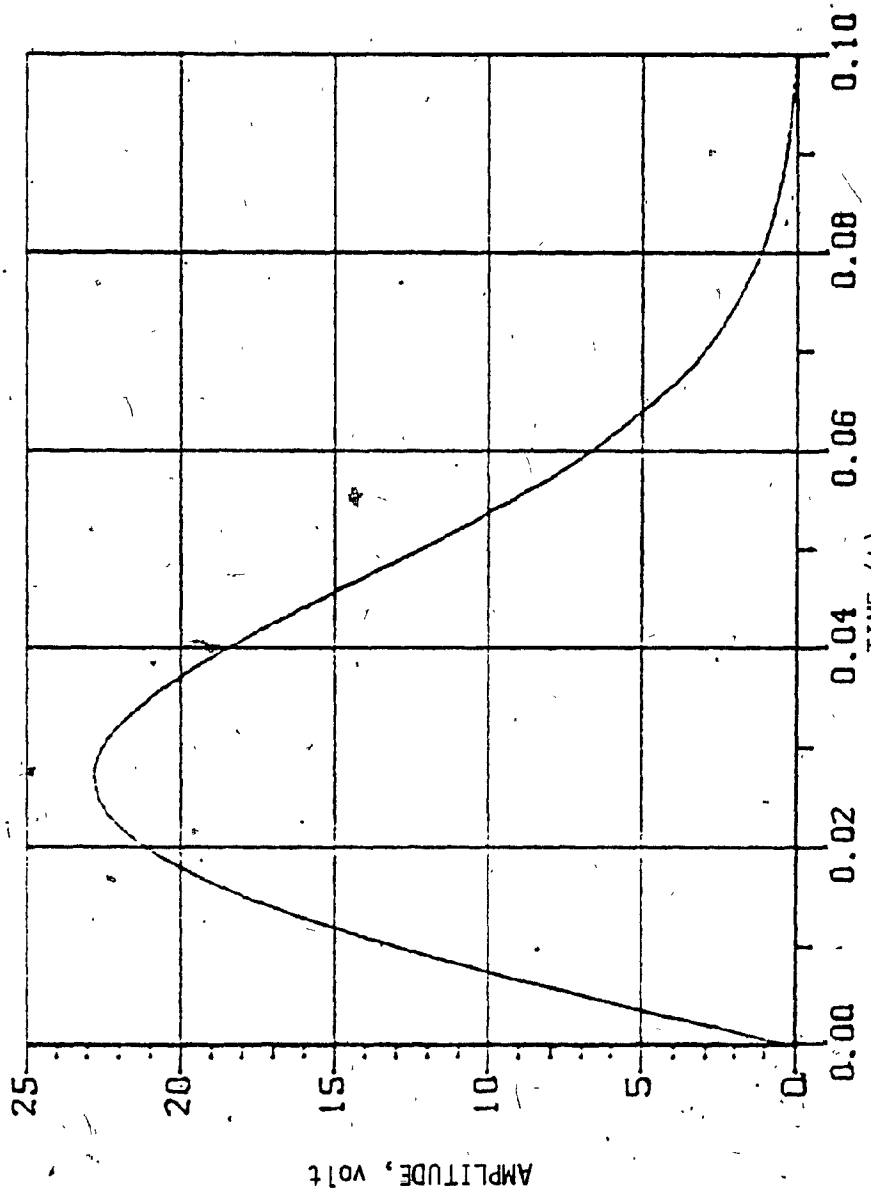


FIGURE 2.10: SMOOTH BOTTOM FUNCTION FOR Φ = 15 deg.

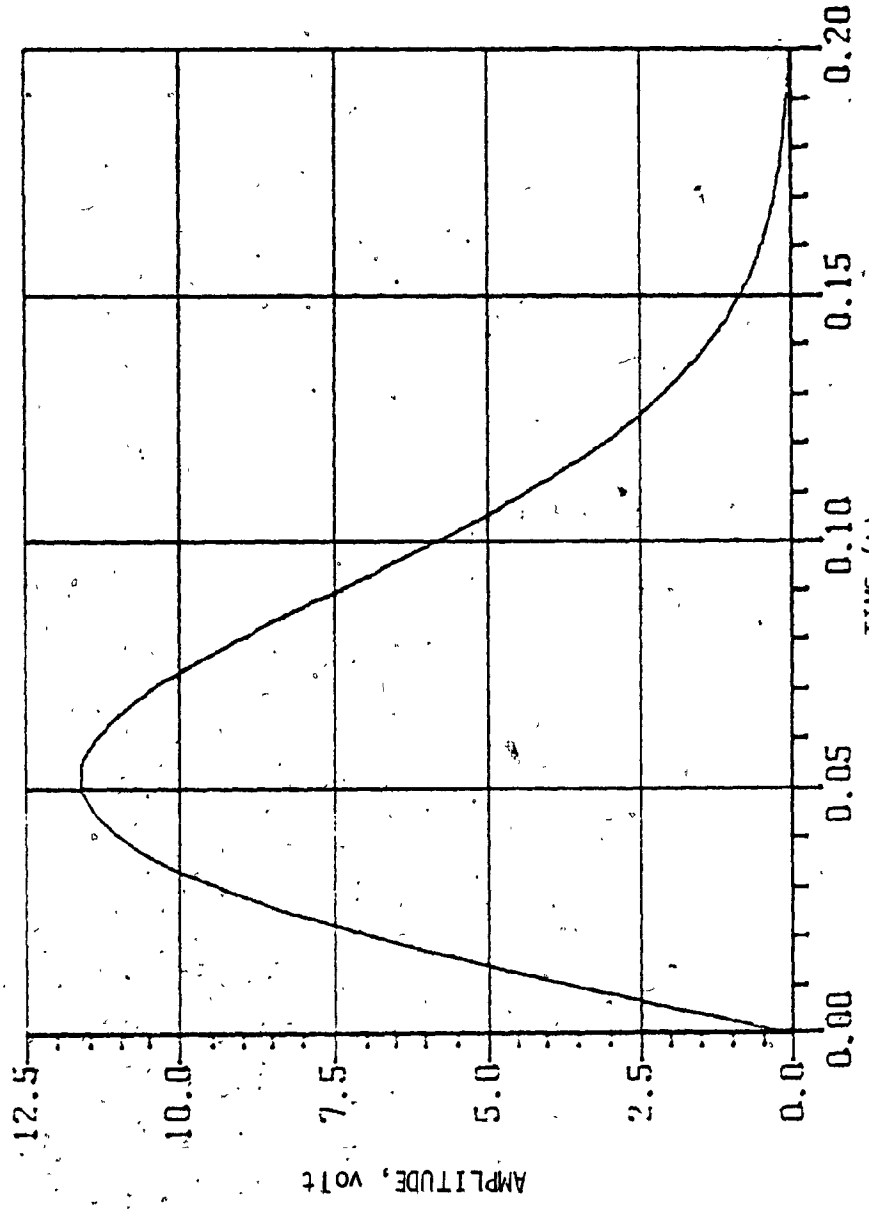


FIGURE 2.11: SMOOTH BOTTOM FUNCTION FOR Φ = 30 deg.

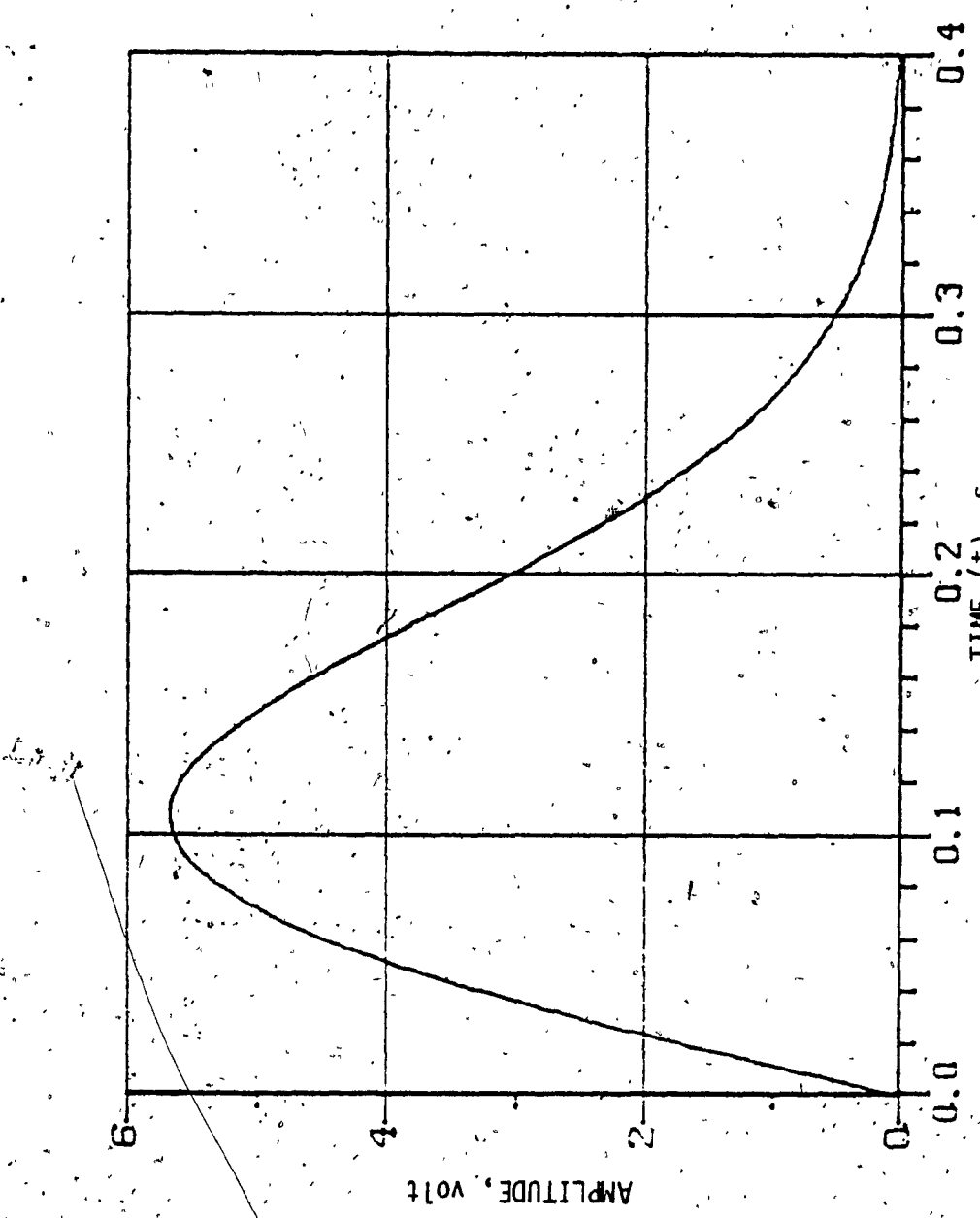


FIGURE 2.12: SMOOTH BOTTOM FUNCTION FOR Φ = 45 deg.

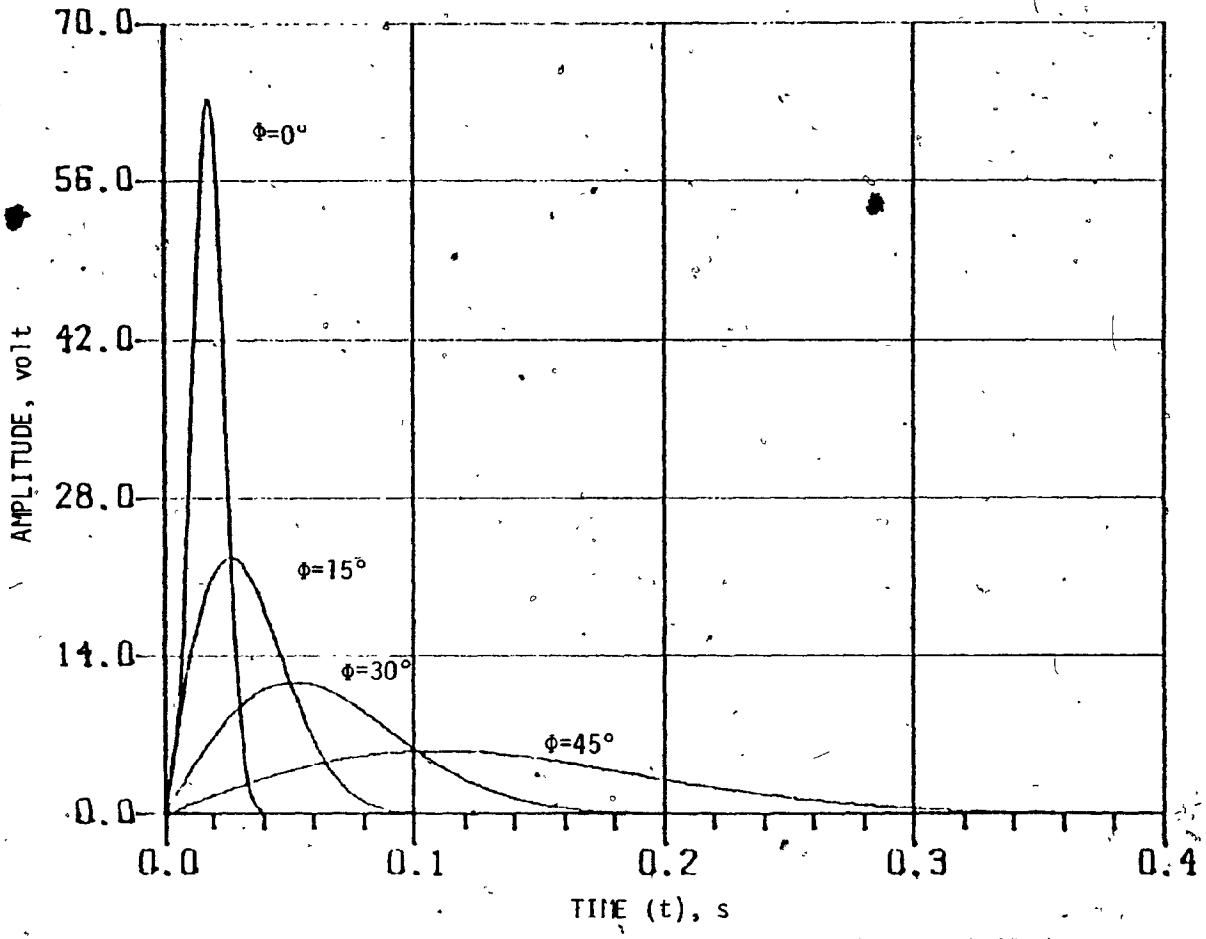


FIGURE 2.13: BOTTOM RETURN SIGNAL ENVELORE FOR Φ = 0-45 deg.

The peak amplitude A_p, the corresponding time t_p, the -3 dB bandwidth, the -40dB bandwidth for each of the steering angle(s) are shown in Table 2.2. The expected echo duration $T(\Phi,d)$ for each of the steering angles is seen to correspond closely to the -3dB widths of the bottom return signals.

Once the smooth bottom function is realized, the additive and the multiplicative noise disturbances are introduced to obtain the complete model. Random number generators are used to derive $n_1(k)$ and $n_2(k)$, $k = 1, 2, \ldots K$, the statistically independent white noise sequences normally distributed with mean zero and variances σ_1^2 and σ_2^2, respectively. The noise sequences are passed through digital lowpass filters f_1, f_2 to bandlimit both the noise disturbances.

A third order recursive filter using Butterworth Approximation is used as the lowpass filters f_1 and f_2. The design procedure for the filters is given in Appendix A. The bandwidths of the filters f_1 and f_2 for each steering angle are chosen to obtain a realistic value of + 15 dB, input SNR. The average signal-to-noise ratio (SNR) is given by

$$SNR \simeq 20 \log \frac{\sigma_1}{\sigma_2} \sqrt{\frac{W_1}{W_2}} \qquad (2.35)$$

where W_1, W_2 are the bandwidths of the digital lowpass filter used to bandlimit multiplicative and additive noise respectively.

Using the same third order recursive structure, the filter coefficients are obtained for each of the steering angle. The sampling rate and the noise bandwidth of the filter for the corresponding steering angles are shown in Table 2.3.

TABLE 2.3

NOISE BANDLIMITING LOWPASS FILTER SPECIFICATIONS

STEERING ANGLE Φ, deg	SAMPLING FREQUENCY f_s, Hz	BANDWIDTH OF LOW PASS FILTER (f_1) FOR MULTI. NOISE W_1, Hz	BANDWIDTH OF LOW PASS FILTER (f_2) FOR ADDITIVE NOISE W_2, Hz	INPUT SNR, dB
0	6960	100	300	15.23
15	2750	100	300	15.23
30	1380	100	300	15.23
45	670	100	300	15.23

For each of the steering angles the output $g_1(k)$, $g_2(k)$ of the digital lowpass filters f_1, f_2 are multiplied and added respectively to the smooth bottom return signal samples $h(k)$ to get the discrete bottom return signal $b(k)$. The bottom return signal with noise for all the steering angles are shown in Figures 2.14-2.17.

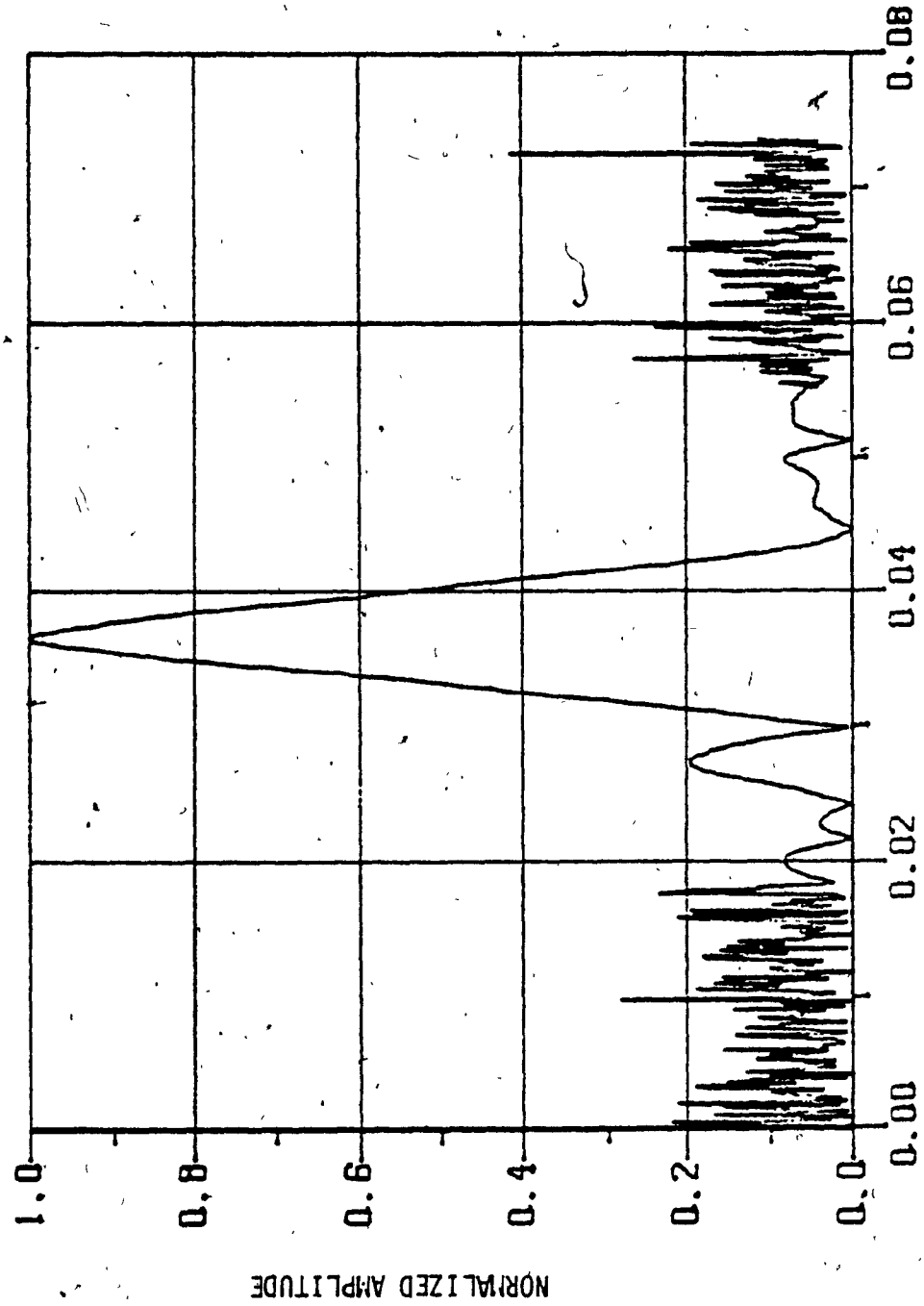


FIGURE 2.14: BOTTOM RETURN SIGNAL FOR Φ = 0 deg.

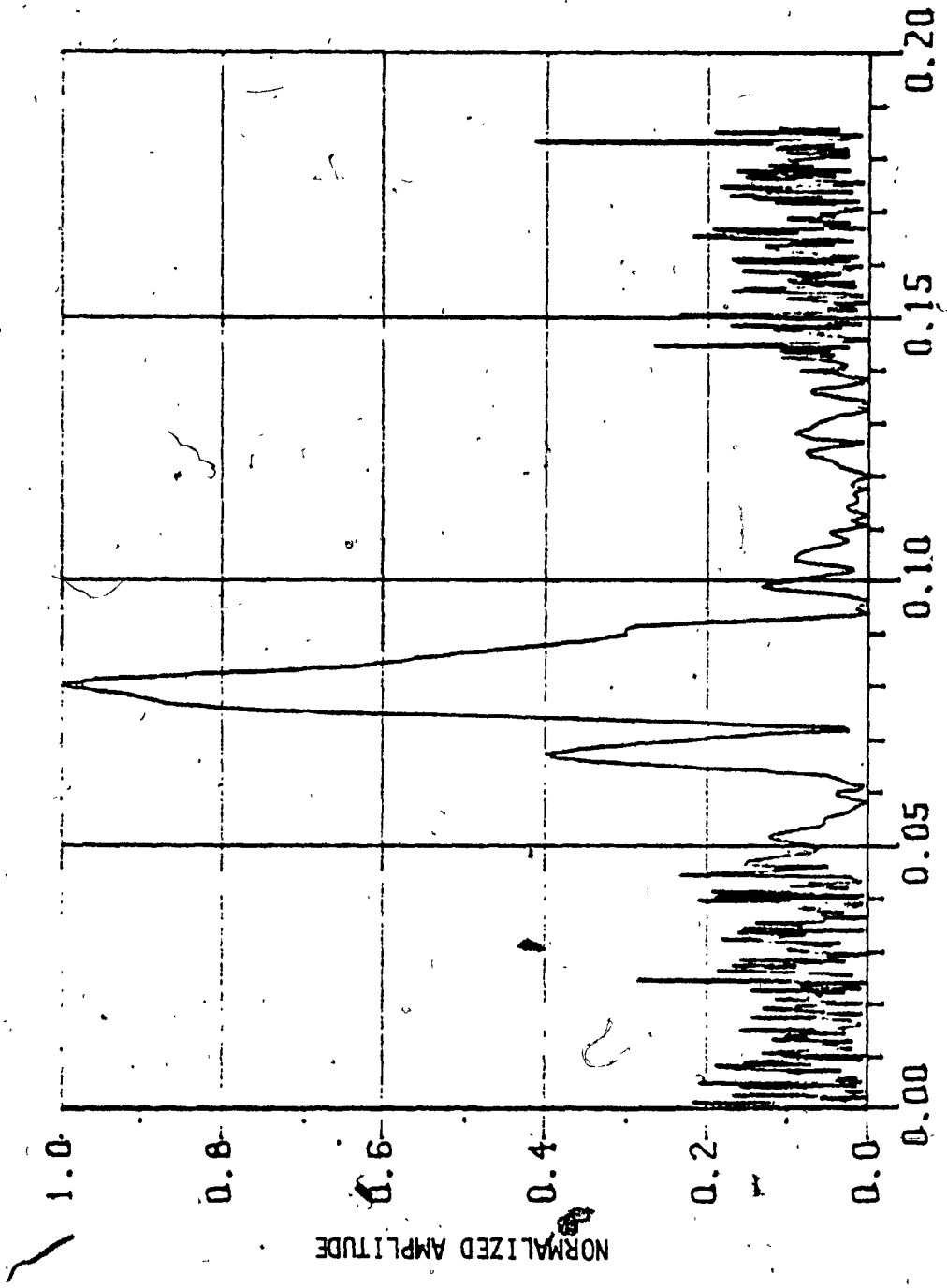


FIGURE 2.15: BOTTOM RETURN SIGNAL FOR Φ = 15 deg.

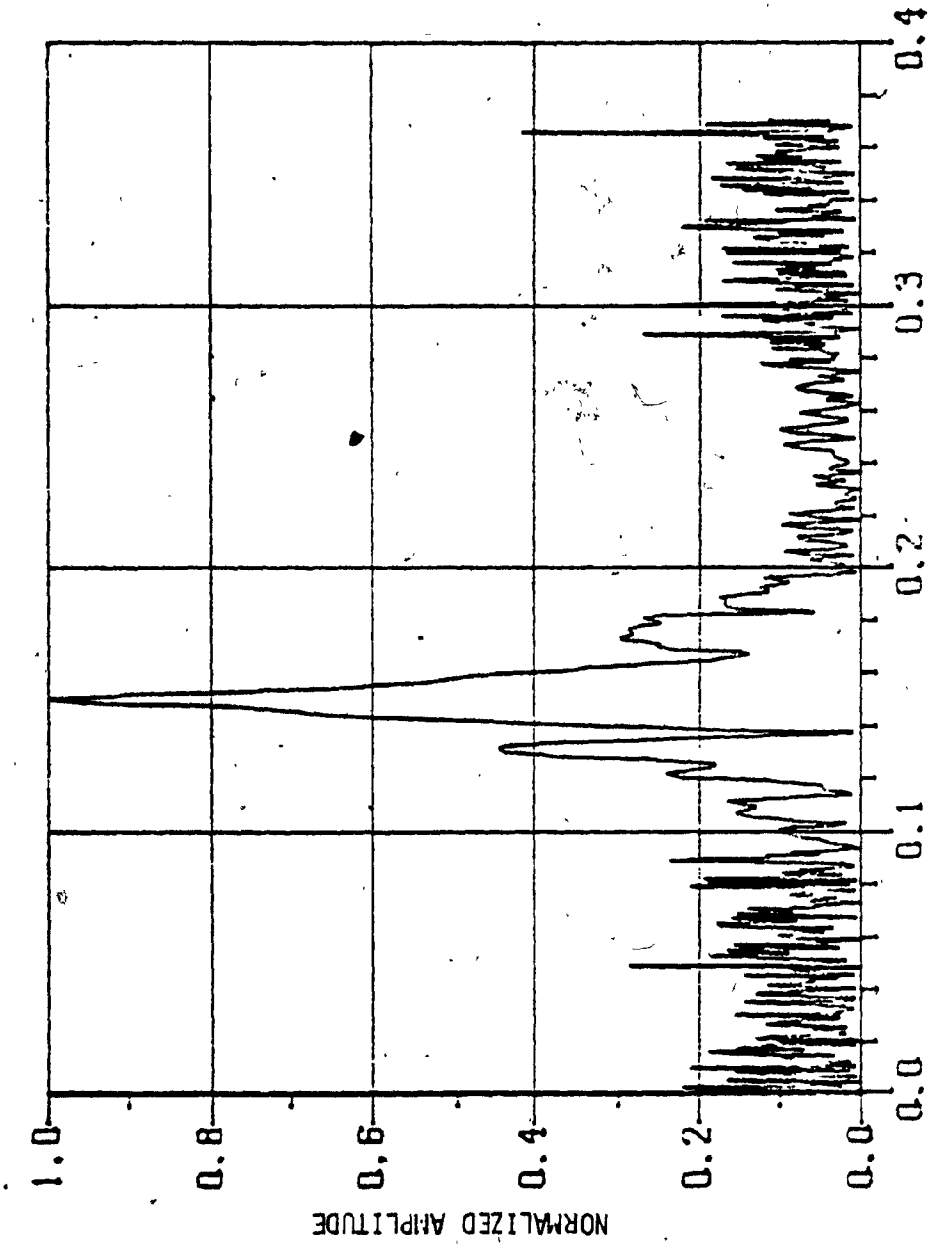


FIGURE 2.16: BOTTOM RETURN SIGNAL FOR Φ = 30 deg.

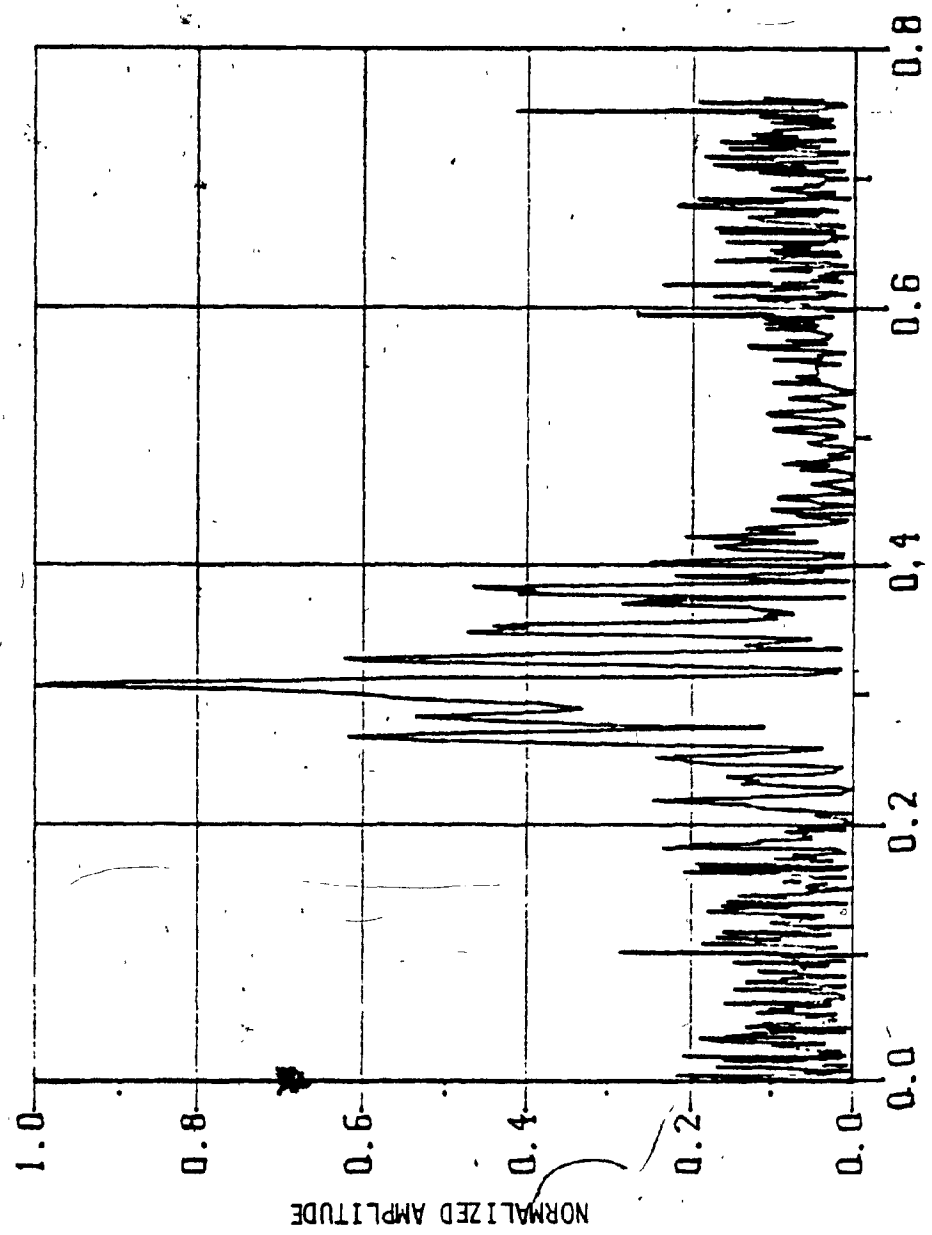


FIGURE 2.17: BOTTOM RETURN SIGNAL FOR Φ = 45 deg.

CHAPTER III

CHAPTER III

DIGITAL SIGNAL PROCESSING TECHNIQUES

3.1 BACKGROUND

In the last chapter a mathematical model was developed for the ocean bottom return signal. The model was simulated using a computer to obtain a realisitc received signal. Once the received signal is obtained, the signal must be processed to extract and/or estimate the values of the desired parameters. In this chapter, a signal processor is developed which makes use of the "a priori" information present in the smooth bottom function to provide a matched filter for the received signal envelope for each spatial direction. The bottom return envelope filter provides a match for the data envelope from each beam $| b(\Phi,d) |$ to the smooth bottom function $h(\Phi,d)$ of the input signal model. An appropriate window (the cosine-tapered window) function is applied to the received signal envelope to reduce the side lobe level at the cost of increasing main lobe width.

3.2 GENERAL STRUCTURE OF A SONAR RECEIVER FOR A WIDE SWATH BOTTOM MAPPING SYSTEM

Figure 3.1 illustrates the general structure of a sonar receiver used for a wide swath bottom mapping system. The received data from the hydrophone arrays for each spatial direction is passed through a beam-forming network. The beam former detects signals coming in on a particular beam [15]. The beam former output for each steering angle is then bandpass filtered with a filter whose center frequency is equal to the carrier frequency. The bandpass filter output is passed through

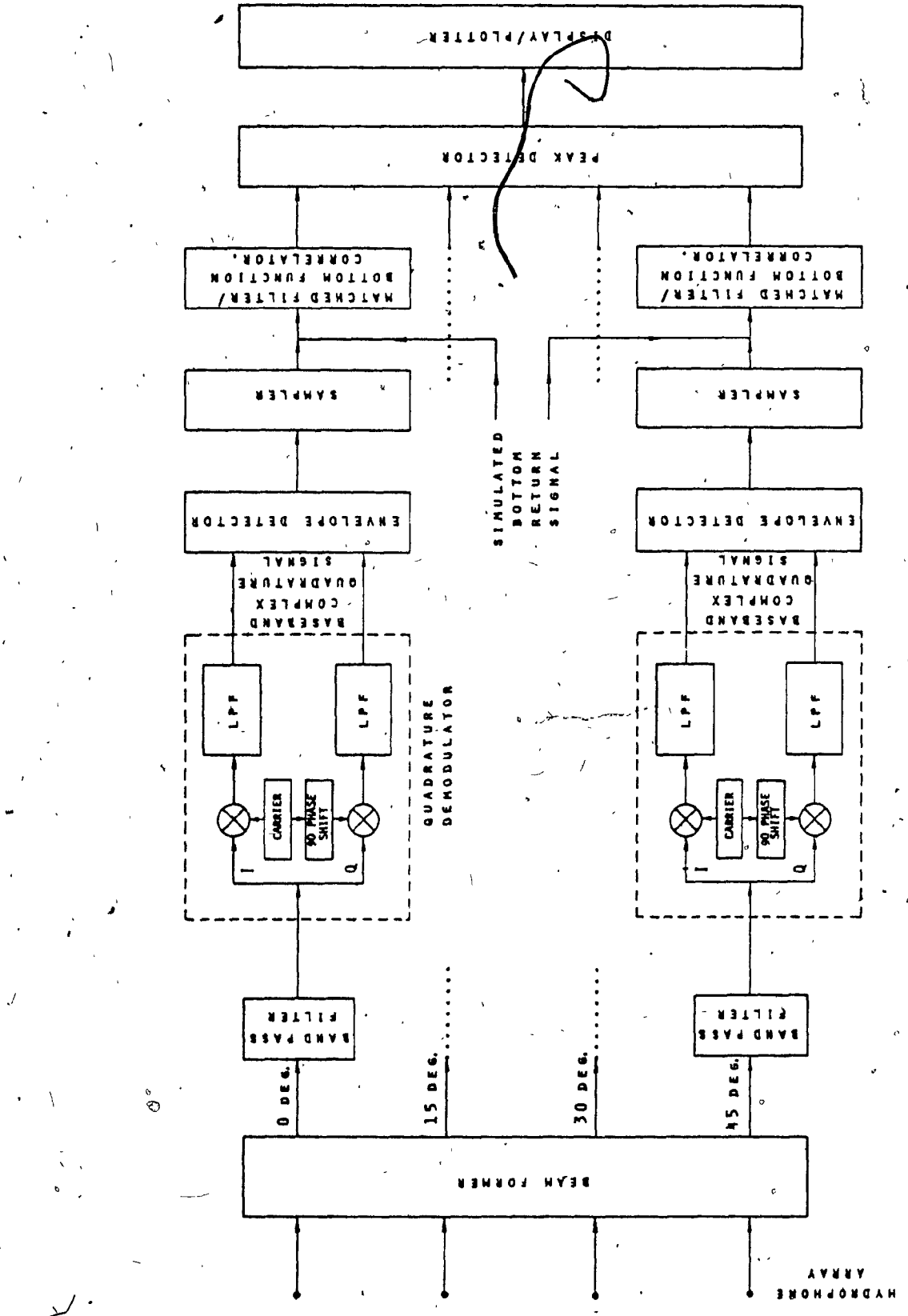


FIGURE 3.1: STRUCTURE OF A SONAR RECEIVER FOR A WIDE SWATH BOTTOM MAPPING SYSTEM

a quadrature demodulator [20,42]. The baseband complex quadrature signal is envelope detected and sampled before feeding into a Digital Matched Filter or bottom function correlator [20,22]. The output of the bottom return matched filter is transferred to a peak detector which determines the peak amplitude point and the corresponding time of the filter output. This time of arrival measurement of the signal determines the slant range. These data are passed on to a display/plotter to obtain the wide swath bathymetric profile. For the purposes of software implementation, the simulated bottom return signal is fed to the matched filter as shown in Figure 3.1 and the output is obtained.

3.3 MATCHED-FILTER THEORY

The matched filter [13,14,22-24] is the filter which optimizes the peak received signal-to-noise ratio by filtering out interfering noise. It is an optimum processor in the sense that it maximizes the probability of detecting the signal in the presence of additive white Gaussian noise [13,14]. So if the signal is present, it peaks at the output of the filter and suppresses the noise amplitude at the same time giving a sharp contrast between the signal and noise.

The matched filter may also be viewed as a form of time-correlator. Consider an input b(t) of finite duration T_w to the matched filter, then its output u(t) is given by

$$u(t) = b(t) * m(t) \qquad (3.1)$$

where m(t) is the impulse response of the matched filter.

Then

$$u(t) = \int_{-\infty}^{\infty} b(\alpha)\, m(t-\alpha)\, d\alpha \qquad (3.2)$$

Now setting the impulse response as

$$m(t) = h(T_w - t) \quad , \quad 0 \leq t \leq T_w \tag{3.3}$$

the output becomes

$$u(t) = \int_{-\infty}^{\infty} b(\alpha)\, h(T_w - t + \alpha)\, d\alpha \tag{3.4}$$

Letting $m(t)$ and $u(t)$ be causal,

$$u(t) = \int_{0}^{t} b(\alpha)\, h(T_w - t + \alpha)\, d\alpha \tag{3.5}$$

Finally, the output sampled at time $t=T_w$ is

$$u(T_w) = \int_{0}^{T_w} b(\alpha)\, h(\alpha)\, d\alpha \tag{3.6}$$

The output $u(T_w)$ of the time correlator can be obtained from the implementation shown in Figure 3.2. From equation (3.3), the impulse response, $m(t)$ of the matched filter is a delayed, time-reversed version of the signal $h(t)$ and is matched to $h(t)$. The optimum matched filter realization is shown in Figure 3.3. It can also be seen that when the filter is matched to $h(t)$, the signal-to-noise ratio is maximized at the output of the matched filter [13-14, 23].

Recalling equations (2.29) and (2.33), $h(t)$ represents the envelope of the received signal (smooth bottom function) for each spatial direction and $b(t)$ is the actual bottom return signal envelope. Hence the impulse response of the matched filter for the data envelope from each beam is a delayed and time reversed version of the smooth bottom function $h(t)$. Ignoring the delay for the time being, the bottom func-

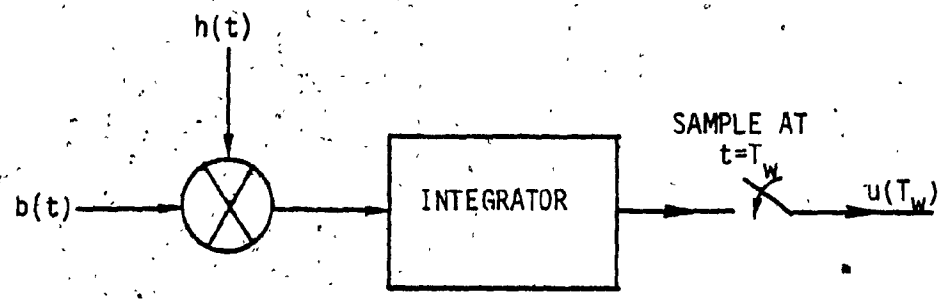


FIGURE 3.2: Correlation Receiver

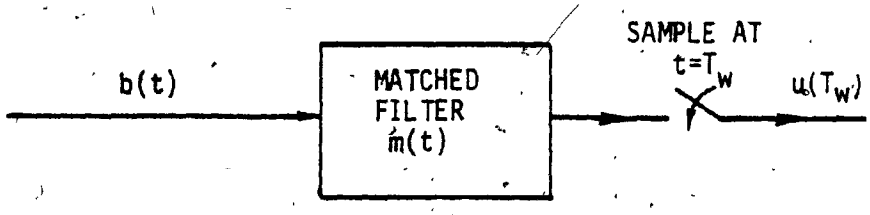


FIGURE 3.3: Matched Filter Realization

tion correlator output is given by

$$u(t) = | b(t) | * h(-t) \qquad (3.7)$$

For the matched filter, the convolution of the received signal envelope with the matched filter impulse response is the same as cross correlation of the received signal envelope with the replica of the signal envelope. Since the shape of the bottom return signal b(t) is dependent on the two-way beam pattern, steering angle (Φ) and depth (d), the bottom return matched filter impulse response should be different for each of the beams comprising the mapping swath. Recollecting from Chapter II, the impulse response of the matched filter, m(t) must be a Gaussian shape for the near vertical steering angles and a Rayleigh shape for the large steering angles, with the -3 dB width very nearly equal to the expected echo duration $T(\Phi,d)$. This close matching of the width is necessary in order to obtain the optimum SNR at the output.

Interpreting equation (3.3) in the frequency domain

$$M(f) = \int_{-\infty}^{\infty} h(T_w - t)\ e^{-j2\pi ft}\, dt$$

$$= \int_{-\infty}^{\infty} h(\alpha)\ e^{-j2\pi f(T_w-\alpha)}\, d\alpha$$

$$= H^*(f)\ e^{-j2\pi fT_w} \qquad (3.8)$$

where * represents complex conjugate.

The exponential in equation (3.8) represents a linear phase shift (bulk delay) to make the filter realizable. Thus, the bottom function correlator output in the frequency domain is

$$U(f) = B(f) \cdot M(f)$$

$$= B(f) \cdot H^{*}(f) \cdot e^{-j2\pi fT_w} \qquad (3.9)$$

Since the matched filter is synthesized using digital implementation, the output of the matched filter given in equation (3.2) can be represented in the discrete form as the convolution sum

$$u(k) = \sum_{n=0}^{K-1} b(n)m(k-n) \;, \quad k = 0,1,2,\ldots,K-1 \qquad (3.10)$$

$$= b(k) * m(k) \qquad (3.11)$$

where

$b(k) = b(kT_s)$ = finite duration data sequence (real) , $k = 0,1,\ldots,K-1$

$m(k) = m(kT_s)$ = finite duration impulse response sequence (real) of the matched filter , $k = 0,1,\ldots,K-1$

From equation (3.3)

$$m(k) = h(-k) \qquad (3.12)$$

Hence equation (3.11) becomes

$$u(k) = b(k) * h(-k) \qquad (3.13)$$

where

$h(k) = h(kT_s)$ = finite duration real valued sequence of the smooth bottom function , $k = 0,1,\ldots,K-1$

The received signal envelope $h(t)$ is of finite duration T_w and is given by

$$T_w = KT_s \qquad (3.14)$$

where K is the number of samples in the signal range $0 \leq t \leq T_w$

(cf. Table 2.2).

3.4 WINDOW SELECTION

3.4.1 Background

A finite duration causal impulse response h(k) is obtained by simply truncating an infinite duration impulse response $h_d(k)$ by multiplying it with a finite duration "window" w(k), i.e.,

$$h(k) = h_d(k)\, w(k) \tag{3.15}$$

where
$$h(k) = \begin{cases} h_d(k), & 0 \le k \le K-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.16}$$

and
$$w(k) = \begin{cases} 1, & 0 \le k \le K-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

This process of obtaining the finite length record by windowing [25-26] is shown in Figure 3.4. In the case where the time signal is abruptly terminated without modification of any coefficients, the window function is rectangular (unweighted). Since any multiplication in the time-domain corresponds to convolution in the frequency-domain, the actual frequency response is the convolution of the desired frequency response and frequency of the window function. The Fourier transform of the rectangular window exhibits the familiar Gibb's oscillations [25,35] and poor convergence of the resulting series. So it is desirable to find a window that brings the impulse response smoothly to zero instead of terminating abruptly, thereby reducing the spectral leakage associated with finite observation intervals.

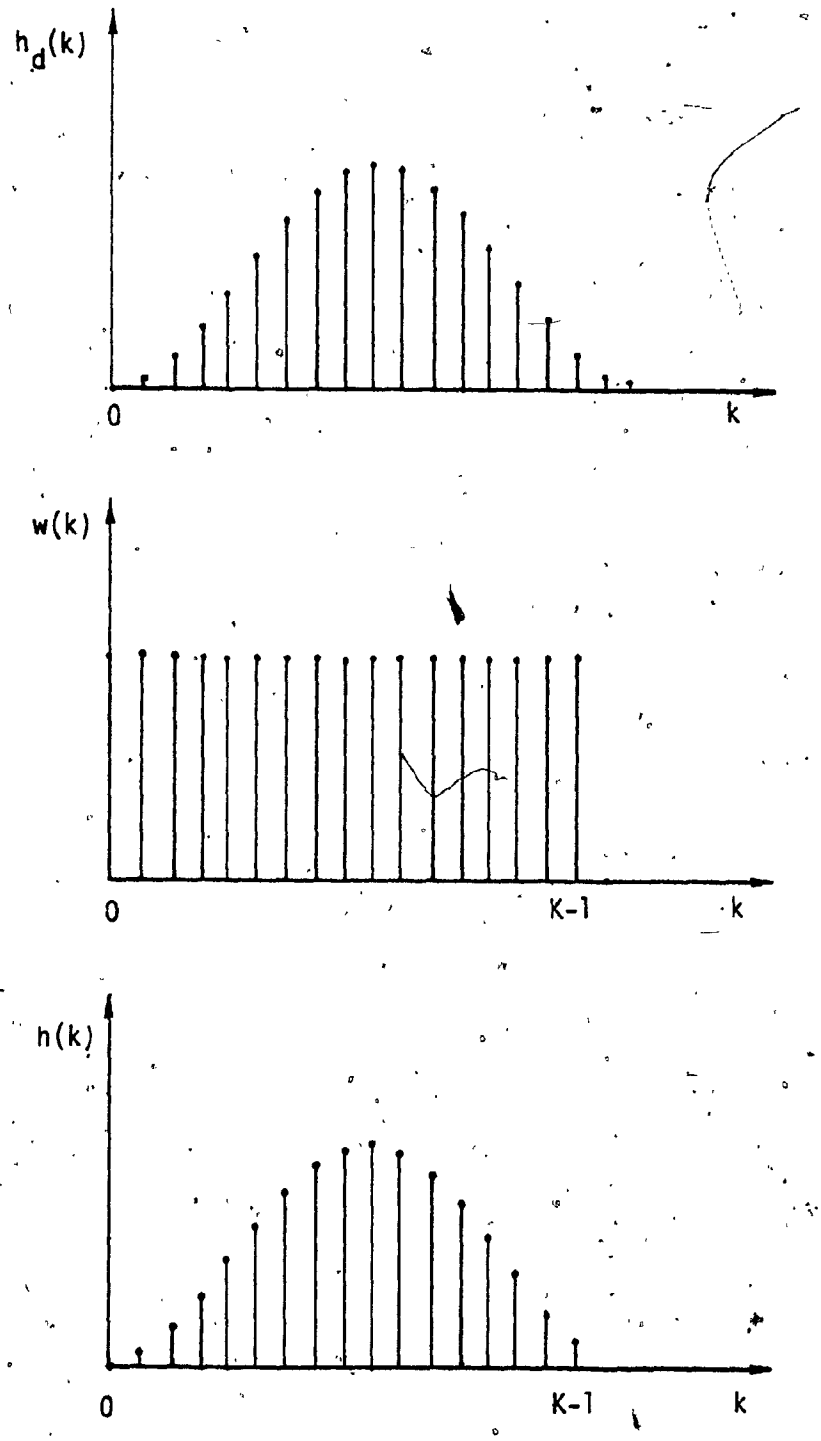


FIGURE 3.4: Equivalent Finite Length Impulse Response Sequence

Since the received signal envelope h(k) for each steering angle is observed for finite duration of T_W seconds, it is equivalent to rectangular windowing. If H(m)# represents the discrete Fourier transform (DFT) of the signal envelope sequence h(k), the transform pair can be stated as

$$H(m) = \sum_{k=0}^{K-1} h(k)\, W_K^{mk} = DFT[h(k)] \qquad (3.18)$$

and

$$h(k) = \frac{1}{K} \sum_{m=0}^{K-1} H(m)\, W_K^{-mk} = IDFT[H(m)] \qquad (3.19)$$

where $W_K = e^{-j(2\pi/K)}$, h(k) and H(m) are complex numbers

The properties of DFT pair is best shown in Figure 3.5. Since the time signal h(k) has K points and is real, there are K/2 unique transform components and the frequency integer m=K/2 corresponds to the folding frequency for the given sampling rate (f_s). If F is the frequency resolution (in Hz) corresponding to the frequency integer (m) increments then it is given by

$$F = 1/T_W = 1/KT_s \qquad (3.20)$$

---

m represents the frequency integer associated with the actual frequency, mF i.e., H(m) = H(mF).

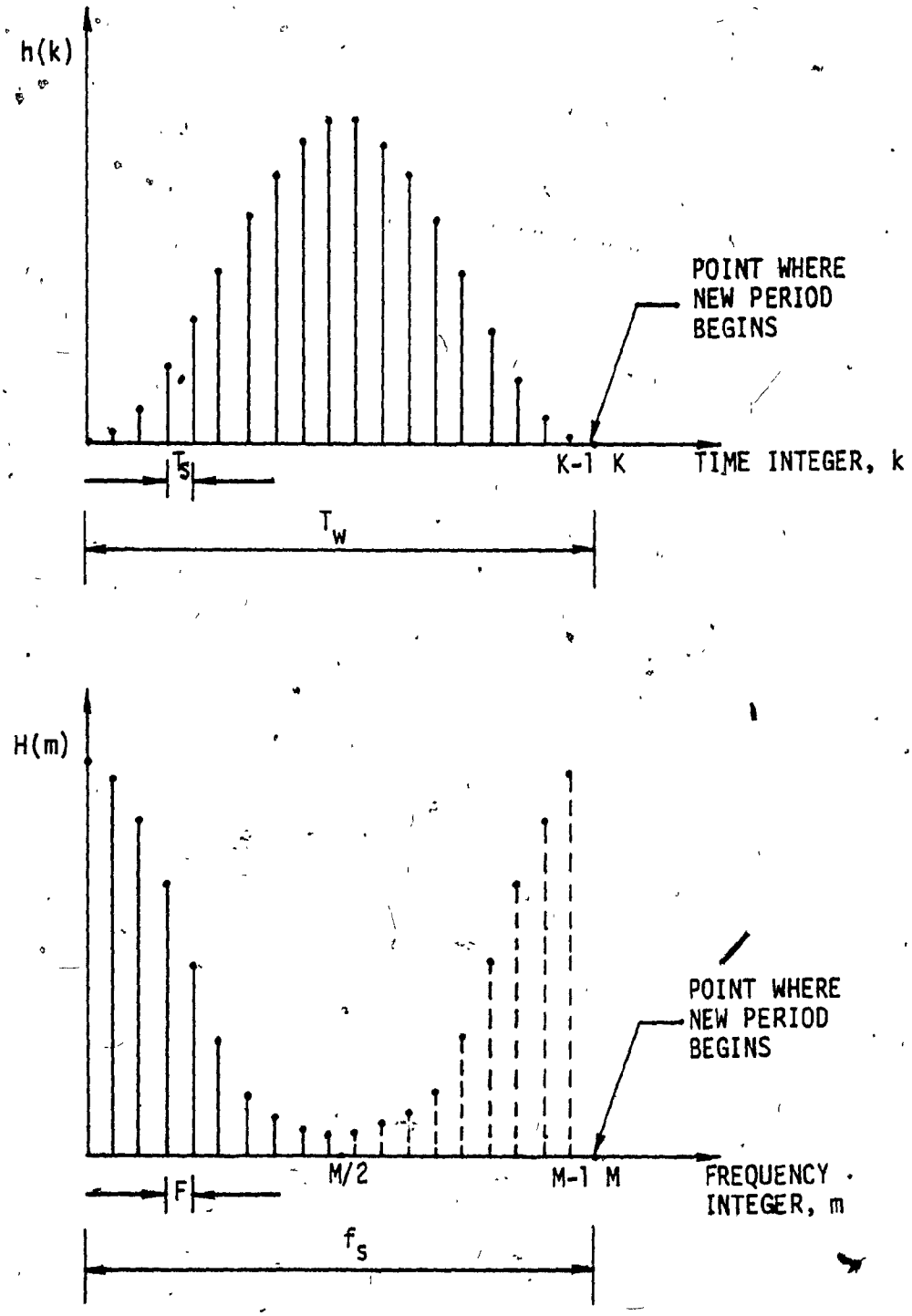


FIGURE 3.5: Properties of DFT Pair

3.4.2 Rectangular Window

This window function is primarily a basis of reference in studying other functions. The K-point Rectangular window corresponds to direct truncation of the discrete smooth bottom function h(k), equivalent to the signal length T_w corresponding to -40 dB width for each of the steering angles. The bottom function is sampled at a rate given in Table 2.2 to obtain the same number of sample points K = 256, for each of the steering angles.

Now the windowed sequence h(k) is given as in equation (3.15) where the Rectangular window is given as

$$w(k) = 1.0 , \quad k = 0,1,\ldots,256. \tag{3.21}$$

Now the magnitude and phase spectrum of h(k) is obtained by obtaining the DFT H(m) for each of the steering angles and is as shown in Figures 3.6 - 3.9.

The programs MODFFT, MFTWW and TUKEY given in Appendix C give the magnitude, phase spectrum and the weighted sequence for different windows considered below. It should be noted that the amplitude spectrum is an even function and the phase spectrum is an odd function due to the properties of DFT and since the sequence h(k) is real, its DFT,

$$H^*(m) = H(-m) = H(k-m) \tag{3.22}$$

The sequence h(k) is interpolated by padding zeros and $N \geq 2K$ point DFT is performed, thereby increasing the frequency resolution. But it is desired to taper the sequence rather than terminate it abruptly by a suitable window function satisfying the two criteria [26-27] :

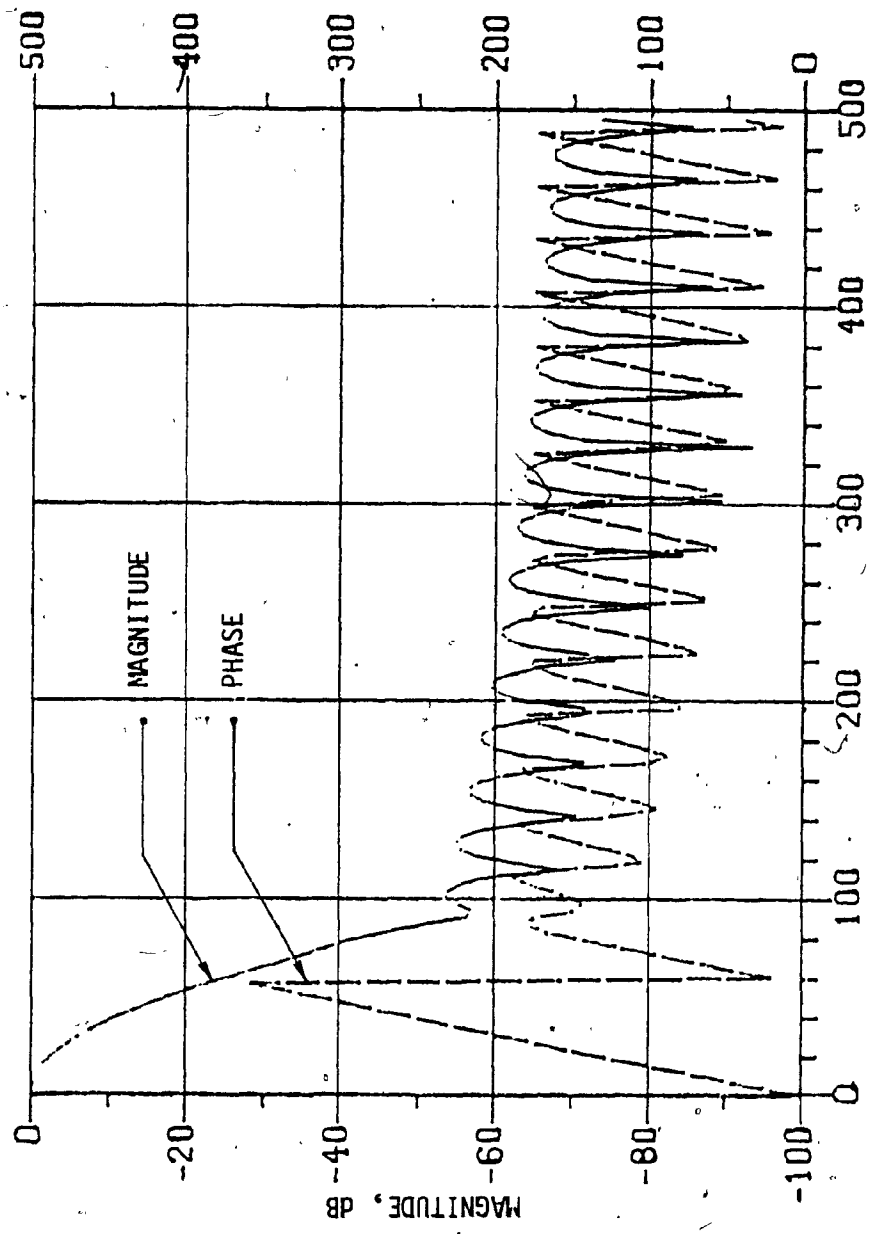


FIGURE 3.6: MAGNITUDE AND PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (UNWEIGHTED) FOR $\phi = 0$ deg.

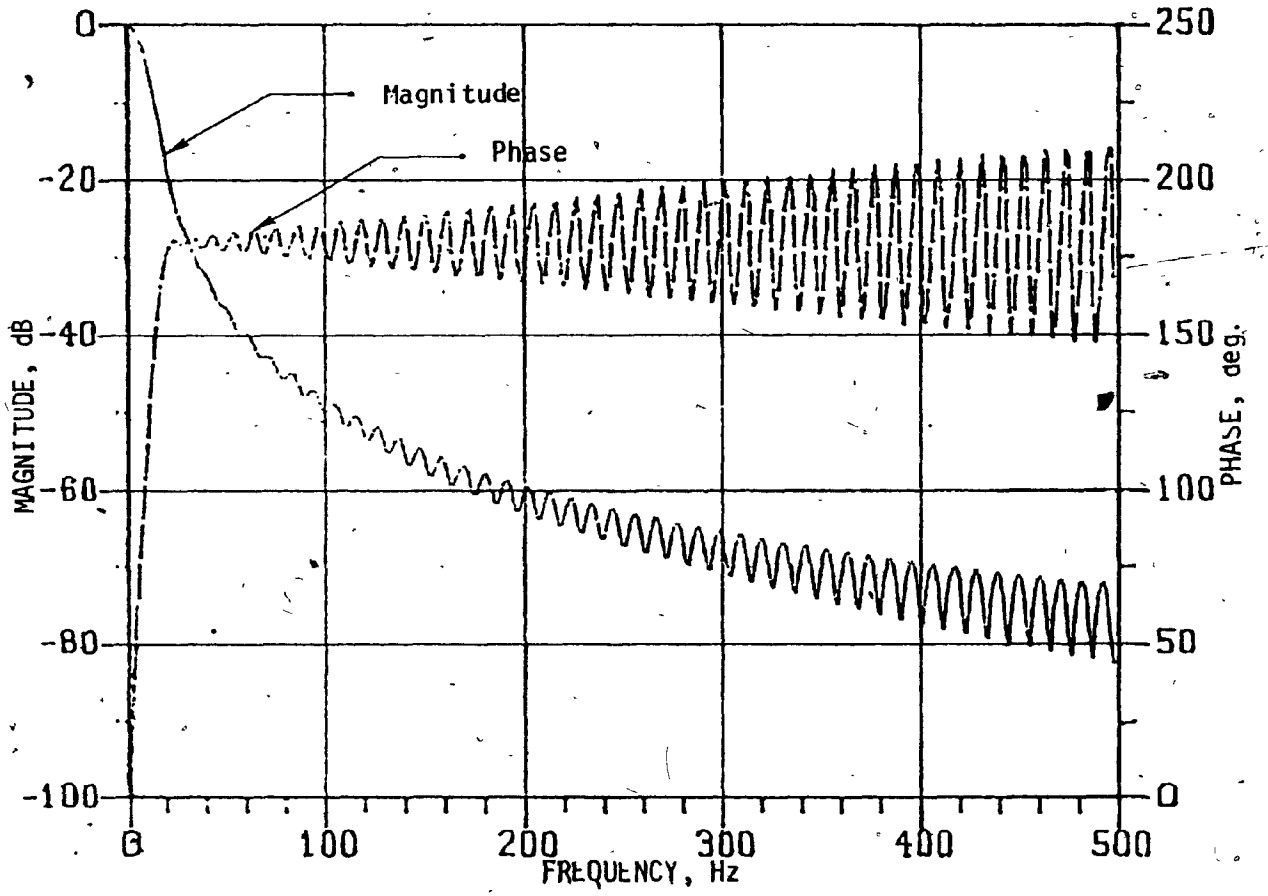


FIGURE 3.7: MAGNITUDE AND PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (UNWEIGHTED) FOR Φ = 15 deg.

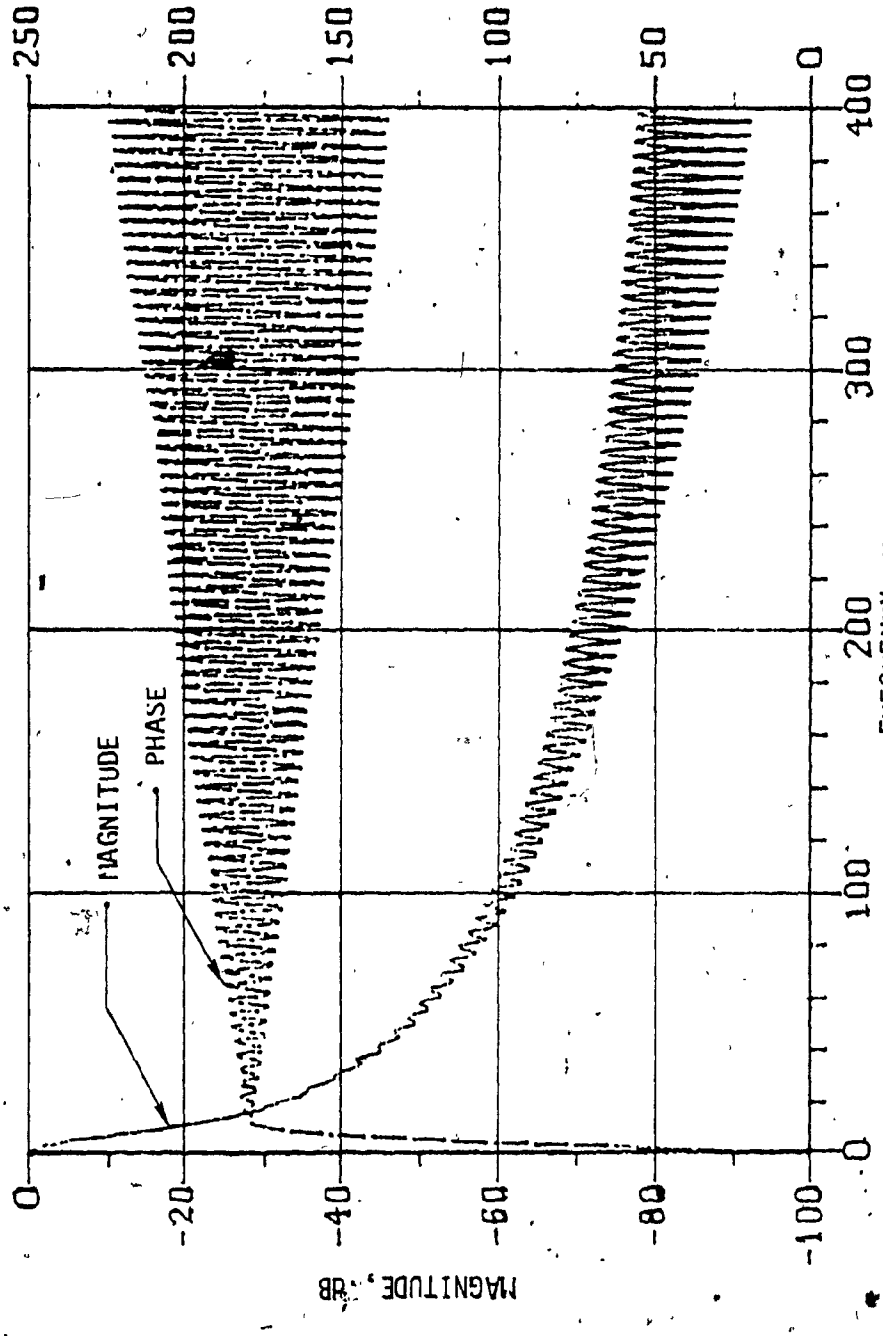


FIGURE 3.8: MAGNITUDE AND PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (UNWEIGHTED) FOR Φ = 30 deg.

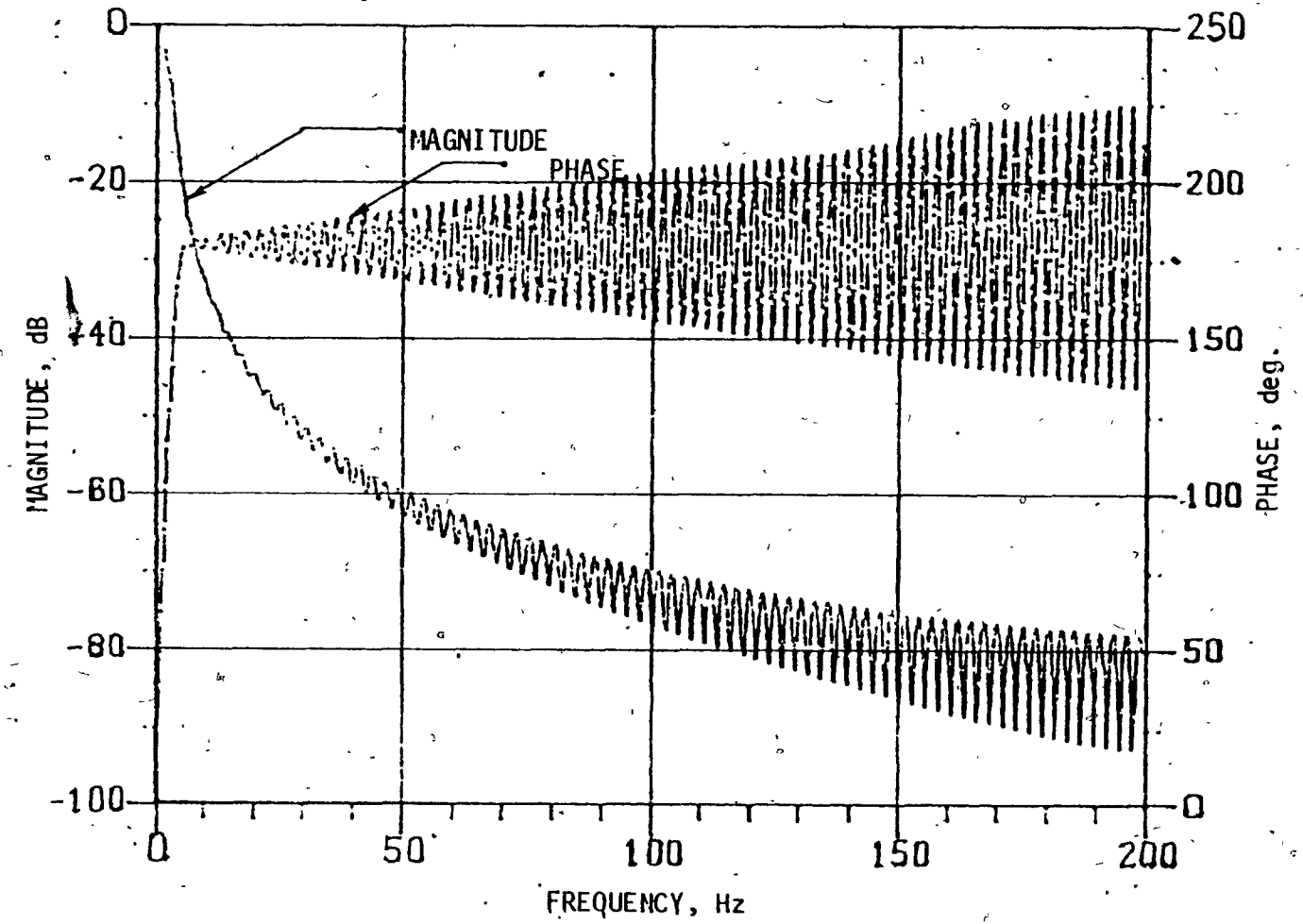


FIGURE 3.9: MAGNITUDE AND PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (UNWEIGHTED) FOR Φ = 45 deg.

(1) Main lobe width (3 dB down) must be equivalent to or as narrow as that of the unweighted spectrum. The -3 dB width of the received signal envelope h(k) must closely match to the expected echo duration in order that the matched filter optimizes the signal-to-noise ratio and is a primary objective in applying the window function.

(2) The maximum side lobe level should be as small as possible relative to the main lobe.

But both these criteria cannot be simultaneously optimized and so a compromise between the two is needed in selecting the appropriate window function.

3.4.3 Frequency Windows [27-29]

Here instead of applying the window as a product in the time domain, it is applied as a convolution in the frequency domain. So we have

$$H_w(m) = H(m) * W(m) \qquad (3.23)$$

where $H(m)$ = frequency spectrum of the finite duration impulse response of the matched filter

$W(m)$ = any frequency window

A. $\cos^{\alpha}(X)$ Windows

This is a family of windows (cosine functions) dependent upon the parameter α, with α being an integer and is given by

$$w(k) = \cos^{\alpha}[\pi k/K] \quad , \quad k = -K/2,\ldots -1,0,1,\ldots,K/2 \qquad (3.24)$$

and for a DFT as

$$w(k) = \sin^{\alpha}[\pi k/K] \quad , \quad k = 0,1\ldots K-1 \qquad (3.25)$$

For $\alpha = 1$, the window is a cosine lobe and equation (3.25) becomes

$$w(k) = \sin [\pi k/K] \quad , \quad k = 0,1,\ldots,K-1 \tag{3.36}$$

Then its windowed sequence $H_w(m)$ is obtained as the convolution in the frequency domain

$$H_w(m) \mid_{\text{cosine}} = 0.5[\, H(m) - H(m+1) \,] \mid_{\text{Rectangle}} \tag{3.37}$$

For $\alpha = 2$, the window is called "Hanning or Hann" [43] and equation (3.35) is represented as

$$\begin{aligned} w(k) &= \sin^2 [\pi k/K] \\ &= 0.5[1.0 - \cos(2\pi k/K)] \quad , \quad k = 0,1,\ldots,K-1 \end{aligned} \tag{3.38}$$

and then the DFT of the windowed sequence is

$$H_w(m) \mid_{\text{Hann}} = 0.5[H(m)] - 0.25[H(m-1) + H(m+1)] \mid_{\text{Rectangle}} \tag{3.39}$$

From equations (3.37) and (3.39) it is seen that the cosine functions have attractive properties when windowed using DFT, and is explained in Appendix B [27-30].

B. HAMMING WINDOW

This is a modified Hann window and is given by

$$w(k) = \alpha - (1-\alpha) \cos[2\pi k/K] \quad , \quad k = 0,1,\ldots,K-1 \tag{3.40}$$

where $\alpha = 0.54$ is selected to have more exact cancellation of the side lobes from the summation of the shifted Dirichlet kernels, i.e.,

$$w(k) = 0.54 - 0.46 \cos [2\pi k/K] \quad , \quad k = 0,1,\ldots,K-1 \tag{3.41}$$

Then the DFT of the windowed sequence is

$$H_w(m) \mid_{\text{Hamming}} = 0.54 [H(m)] - 0.23 [H(m-1) + H(m+1)] \mid_{\text{Rectangle}} \tag{3.42}$$

so the frequency-windowed spectral points are obtained from the rectangle-windowed spectral points by multiplying them with the corresponding

weighting coeficients as given in equations (3.37), (3.39) and (3.42).

3.4.4 Tukey Window

The Tukey window [27,31] also called split cosine-tapered window is a smooth approximation to the Rectangle window. This window smoothly sets the data to zero at the ends without significantly reducing the -3 dB width by leaving the bulk of the data unmodified. The window evolves from the rectangle to the Hanning window as the parameter α_t varies from zero to unity. Thus the window is obtained by splitting the cosine bell into two and inserting a stretch of one's. The window is given by

$$w(k) = \begin{cases} 0.5\,[\,1 - \cos(\pi k/m)\,] & , k = 0,1,\ldots,m-1 \\ 1 & , k = m,\ldots,K-m-1 \\ 0.5\,[\,1 - \cos(\pi(K-k)/m)\,] & , k = K-m,\ldots,K-1 \end{cases} \quad (3.43)$$

where K = 256, the length of envelope sequence to be windowed

$$m = \alpha_t K/2 \quad (3.44)$$

and α_t = the proportion of the data that is tapered (varies from 0 to 1). The Tukey window is applied in time domain so that the windowed sequence is given by

$$h_w(k) = h(k) \cdot w(k) \quad (3.45)$$

where w(k) is the Tukey window sequence and is shown in Figure 3.10. Then the transform of the windowed sequence $h_w(k)$ is

$$H_w(m) = \text{DFT}\,[h_w(k)]\,|_{\text{Tukey}} \quad (3.46)$$

which gives the magnitude spectrum.

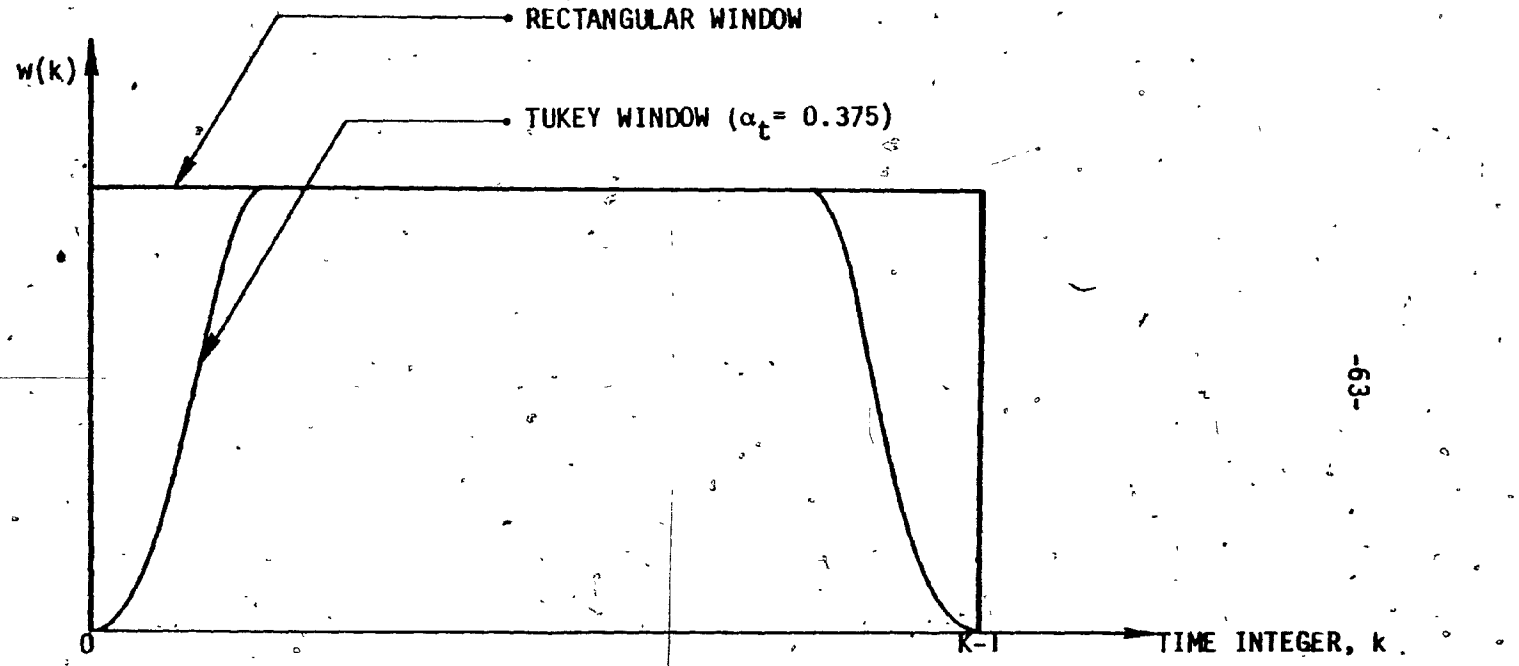


FIGURE 3.10: Tukey Window

3.4.5 Performance Comparison of Windows

Figures 3.11 - 3.30 give the magnitude spectrum of cosine, Hann, Hamming and Tukey window weighted finite duration received signal envelope sequences for each steering angle with the corresponding spectrum of the unweighted sequences (Rectangular window).

It is seen from the transform (Figures 3.11 - 3.22) that the cosine, Hamming and Hann have increasing main lobe width, decreased sidelobes in that respective order reflecting the increased smoothness of the window. Since the highest sidelobe level is well below -40 dB for even the unweighted case, the reduction in sidelobe level could be sacrificed for reducing the mainlobe width for near-vertical steering angle.## The Tukey window does exactly this. For other steering angles (ϕ = 15-45 deg.), the sidelobes are not predominant because the signal envelope itself is smoothly approaching zero. For these cases also the minimisation of mainlobe width is more important in the trade-off between narrow mainlobe width and low sidelobes according to the criteria mentioned in the previous section. The Tukey window seems to be the best choice for this application. When α_t,# the parameter that controls the

---

α_t, the proportion of the signal envelope sequence to be tapered at both ends of the sequence can be the same when $\alpha_t = \alpha_{t1} = \alpha_{t2}$ or different when $\alpha_t \neq \alpha_{t1} \neq \alpha_{t2}$, where α_{t1} is for the leading portion and α_{t2} for the trailing portion of the received signal envelope sequence.

It should be noted that the highest side lobe level of the Tukey weighted spectrum is higher than the unweighted spectrum since Tukey window "truncates" bottom return signal envelope at higher amplitude points than Rectangular window.

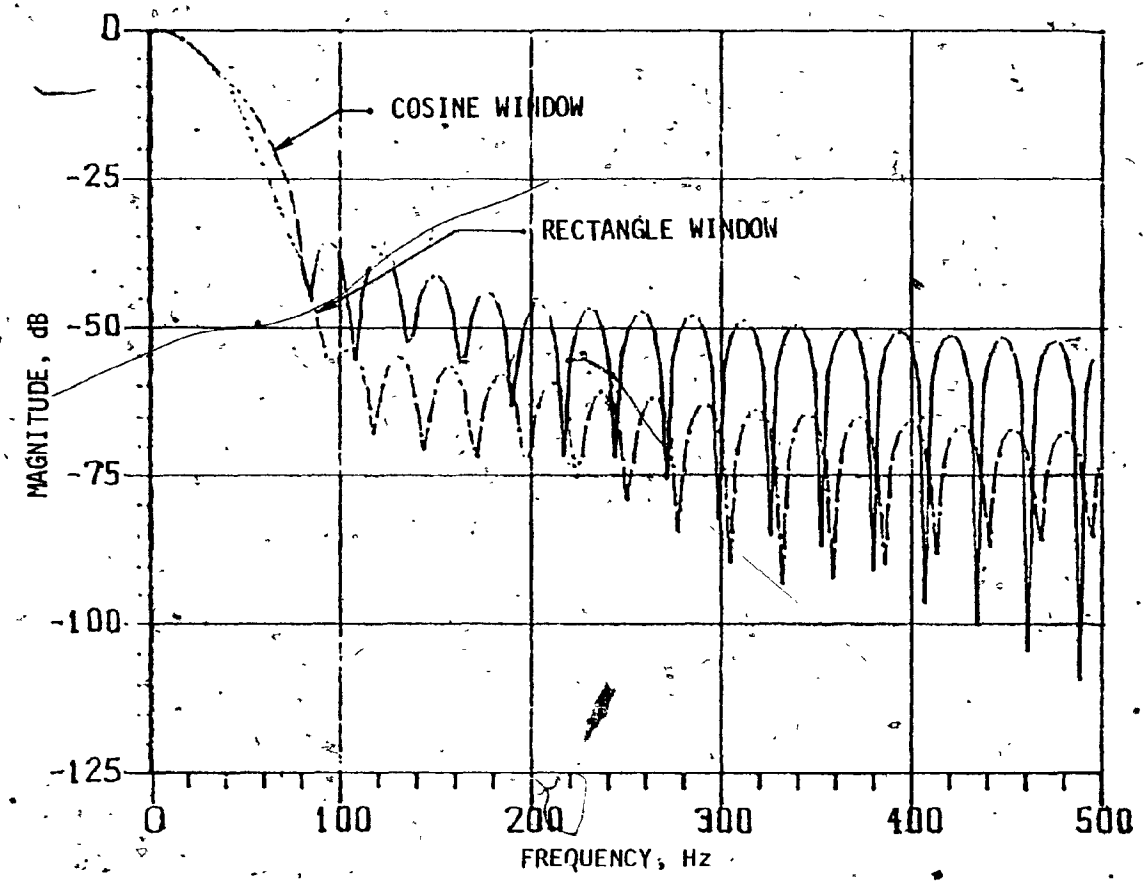


FIGURE 3.11: MAGNITUDE SPECTRUM OF BOTTOM RETURN-SIGNAL ENVELOPE (Weighted) FOR Φ = 0 deg.

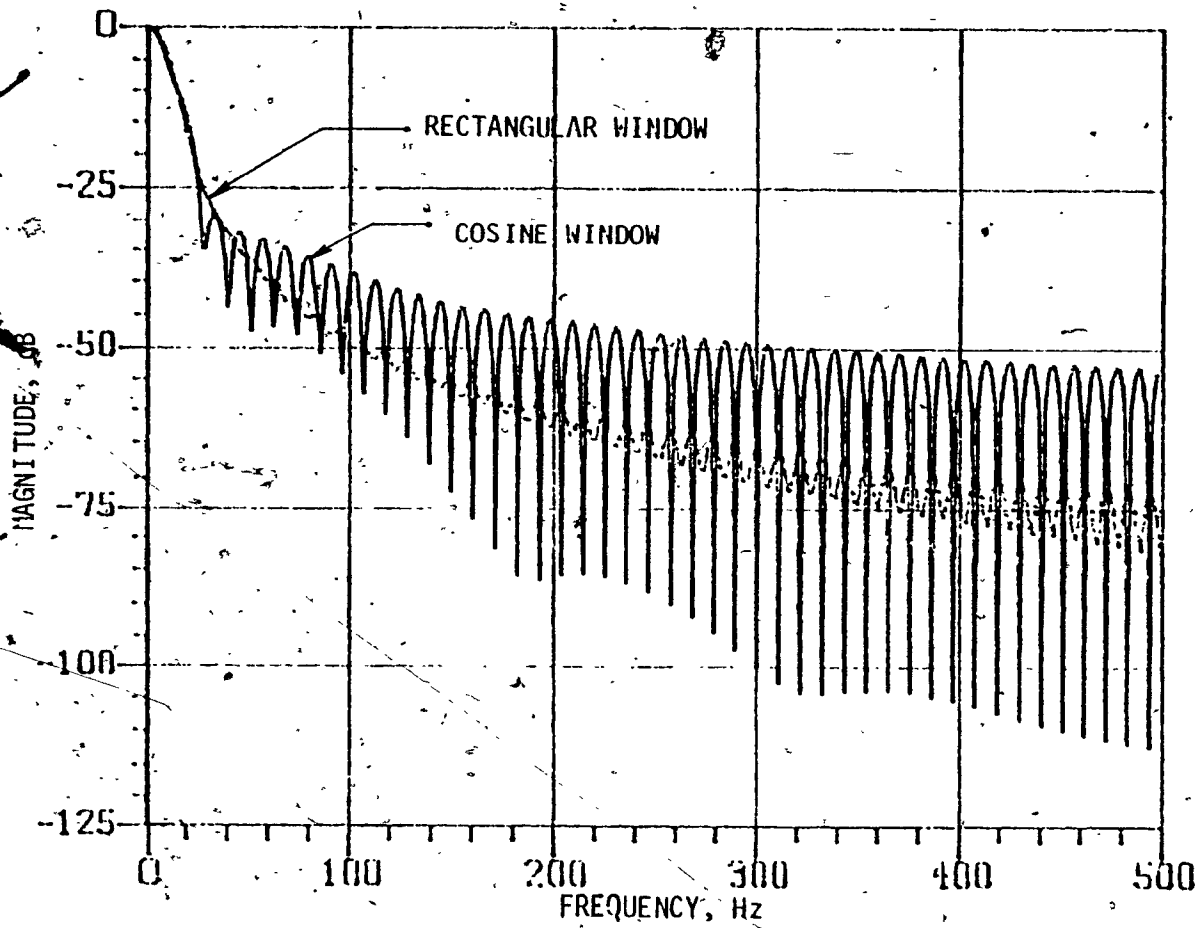


FIGURE 3.12: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 15° deg.

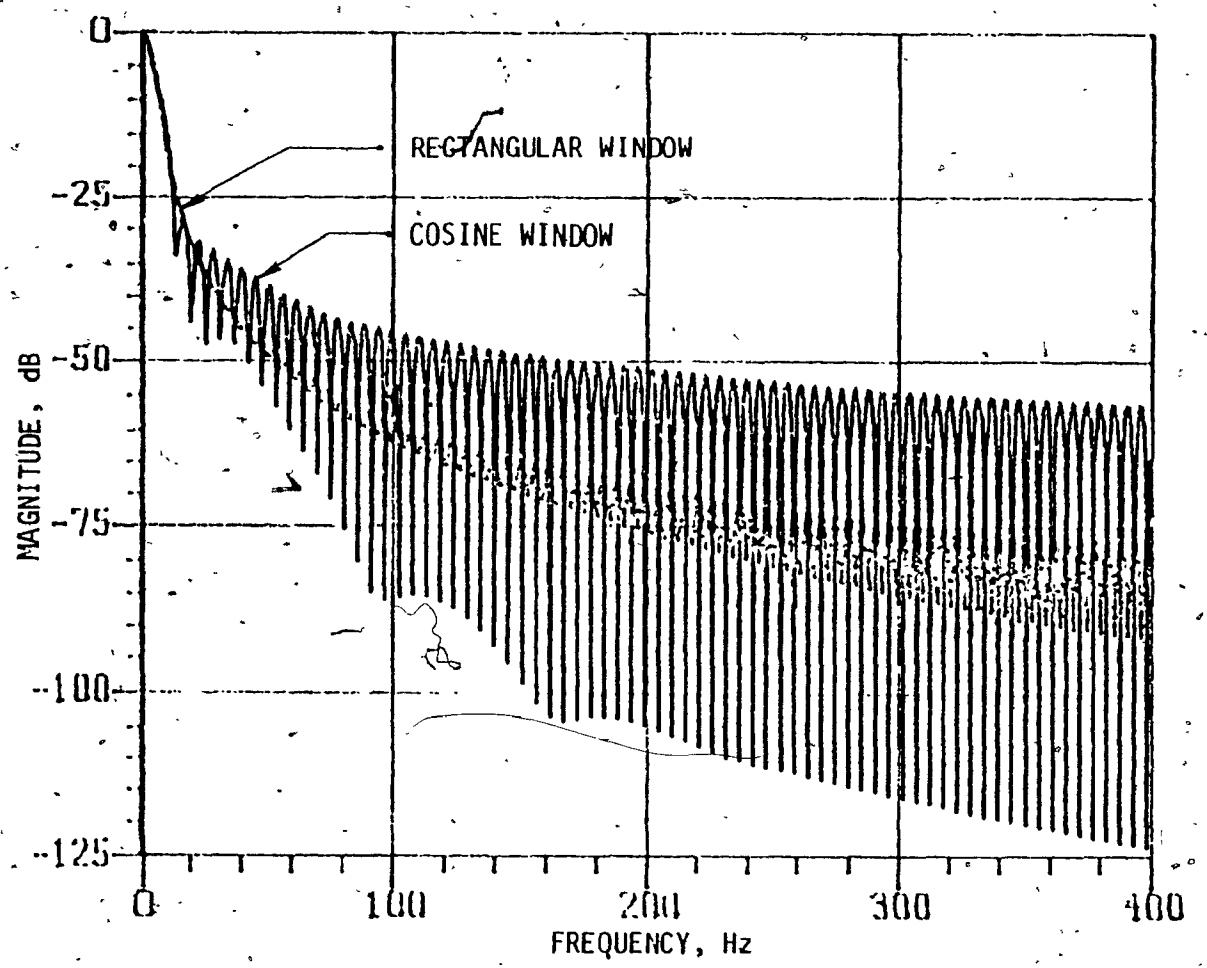


FIGURE 3.13: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 30 deg.

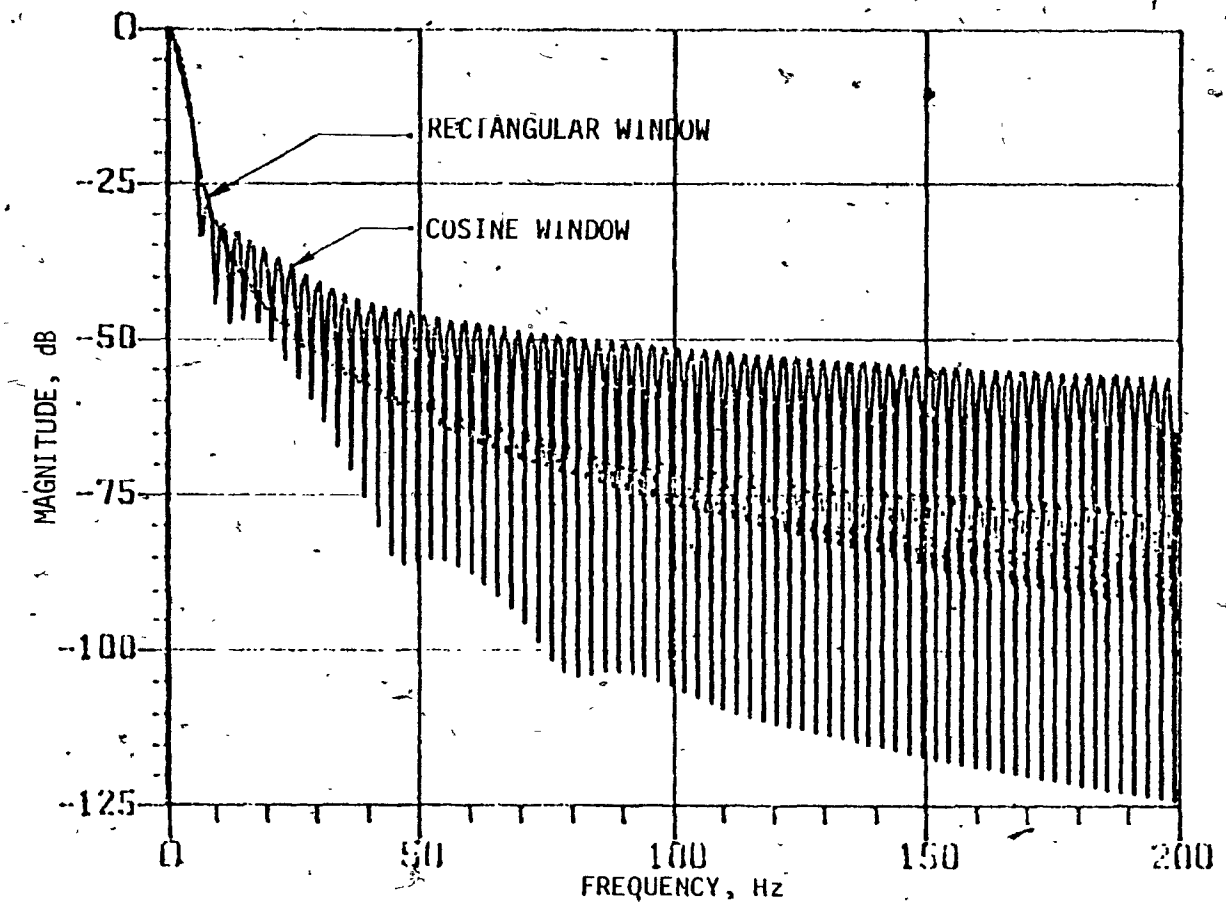


FIGURE 3.14: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 45 deg.

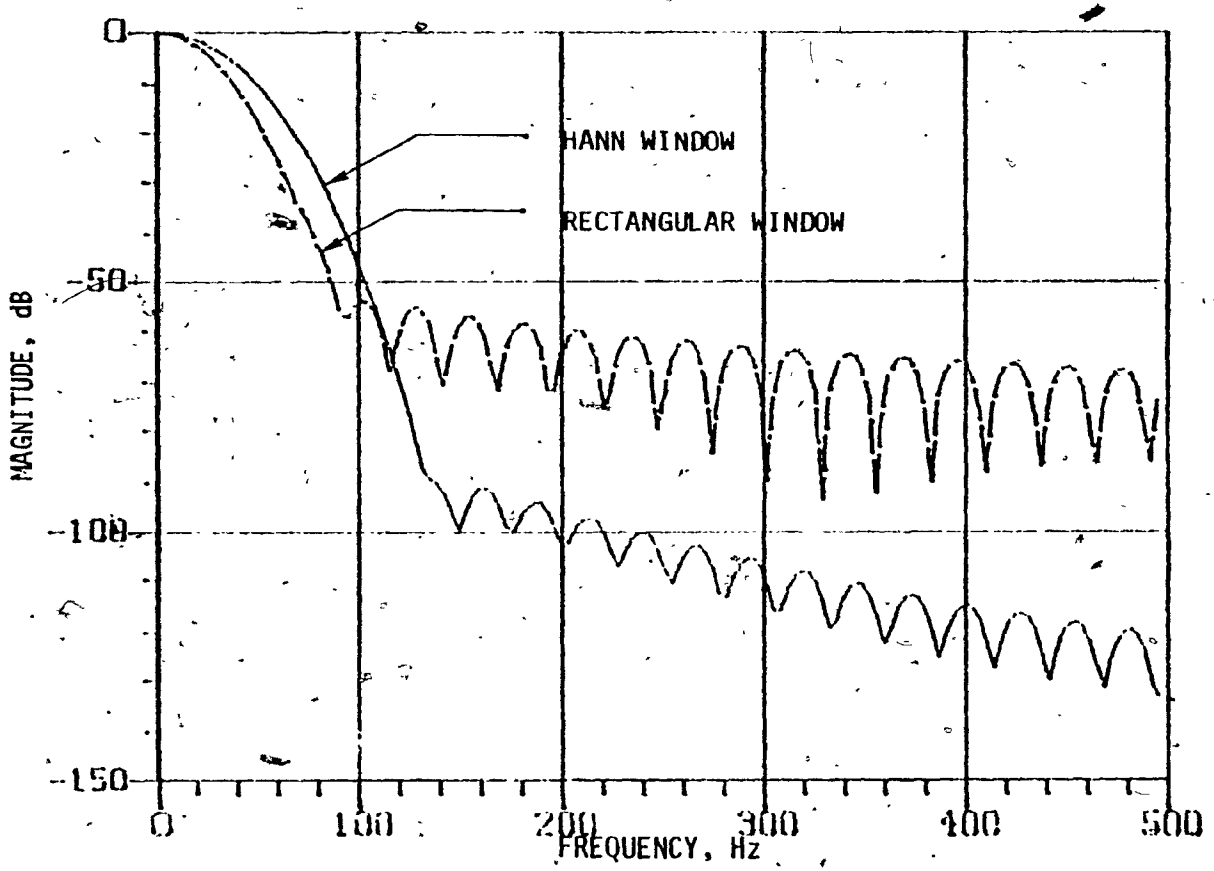


FIGURE 3.15: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 0 deg.

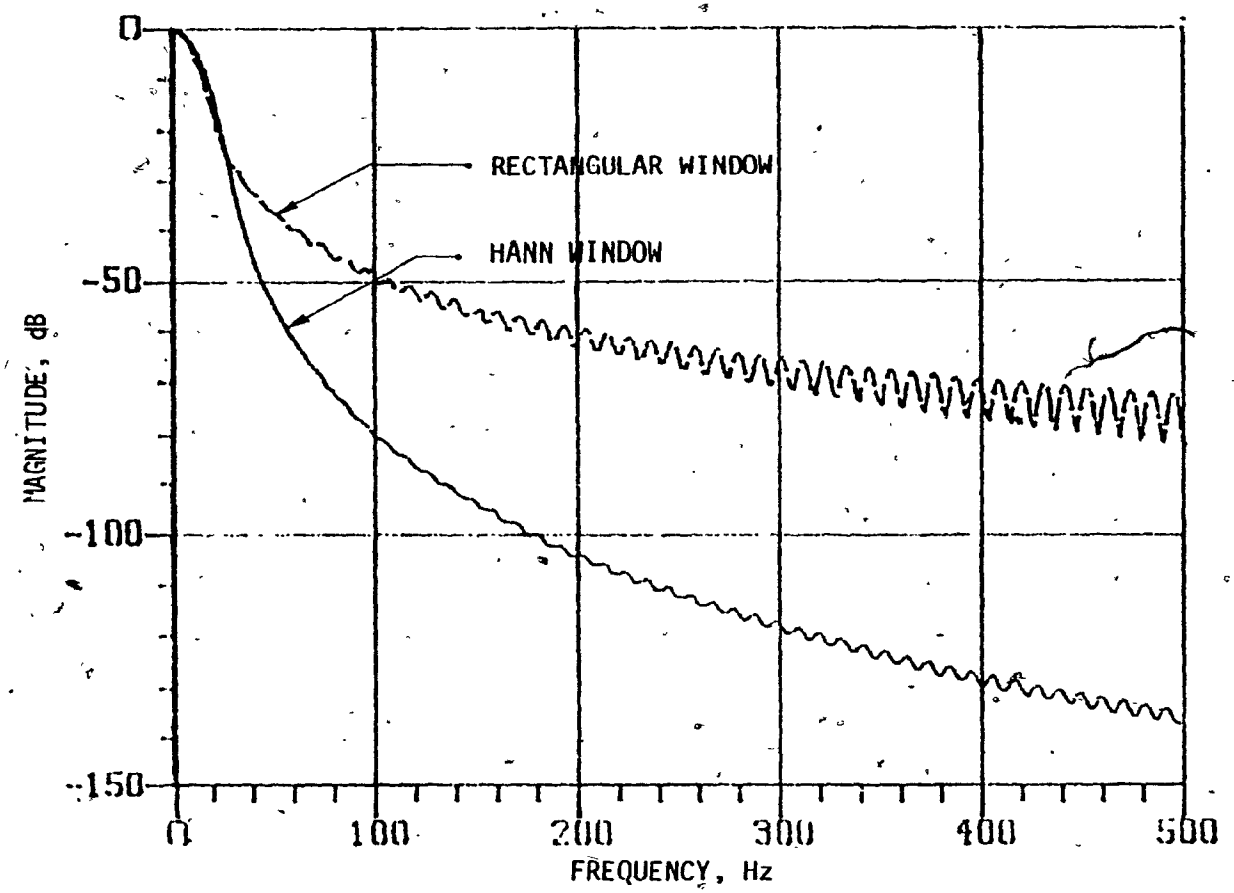


FIGURE 3.16: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR $\bar{\Phi}$ = 15 deg.

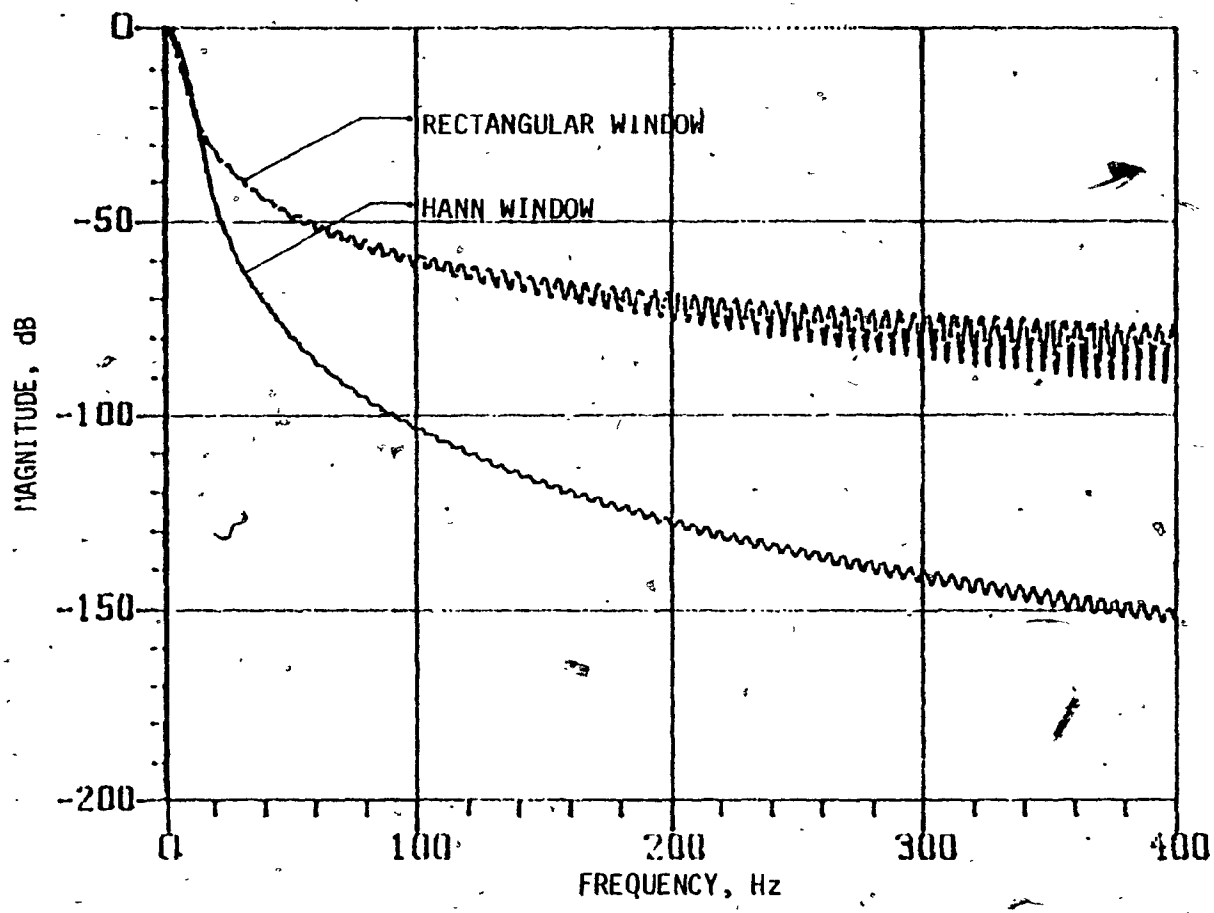


FIGURE 3.17: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 30 deg.

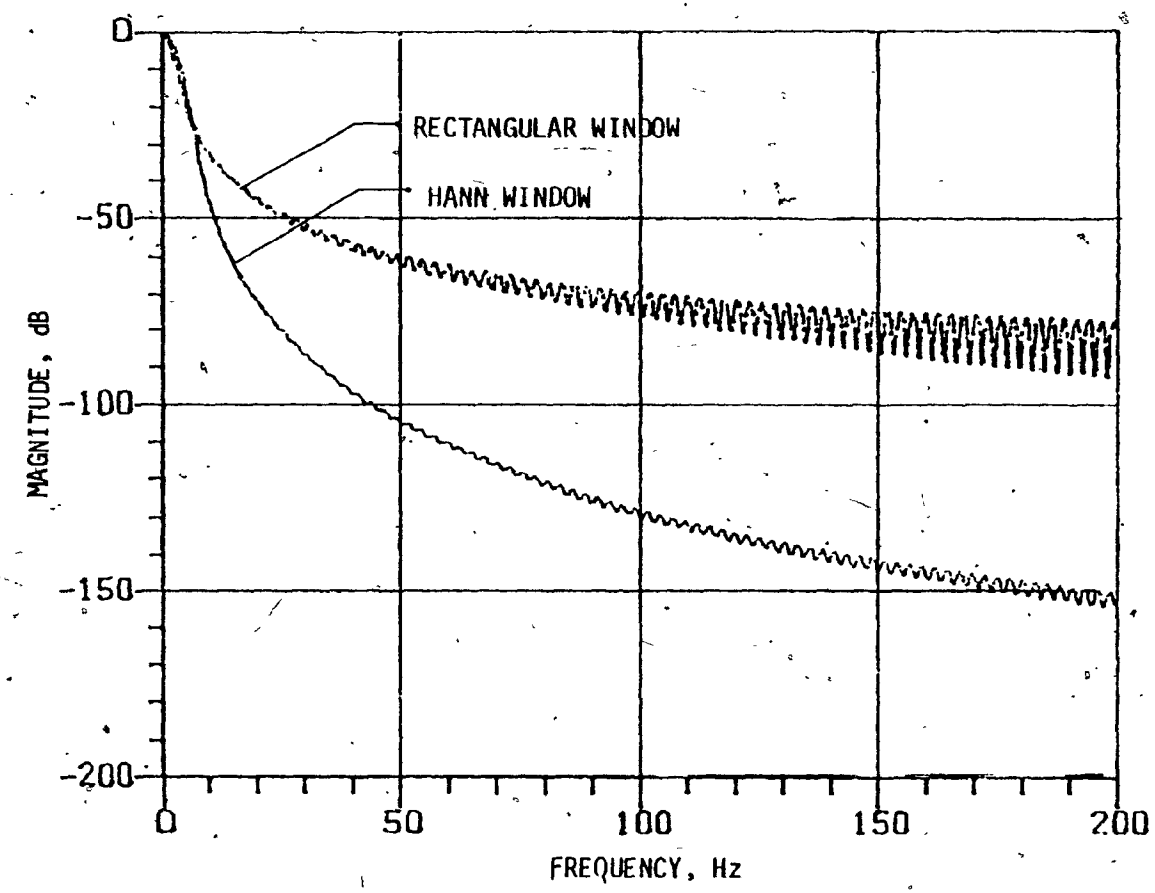


FIGURE 3.18: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 45 deg.

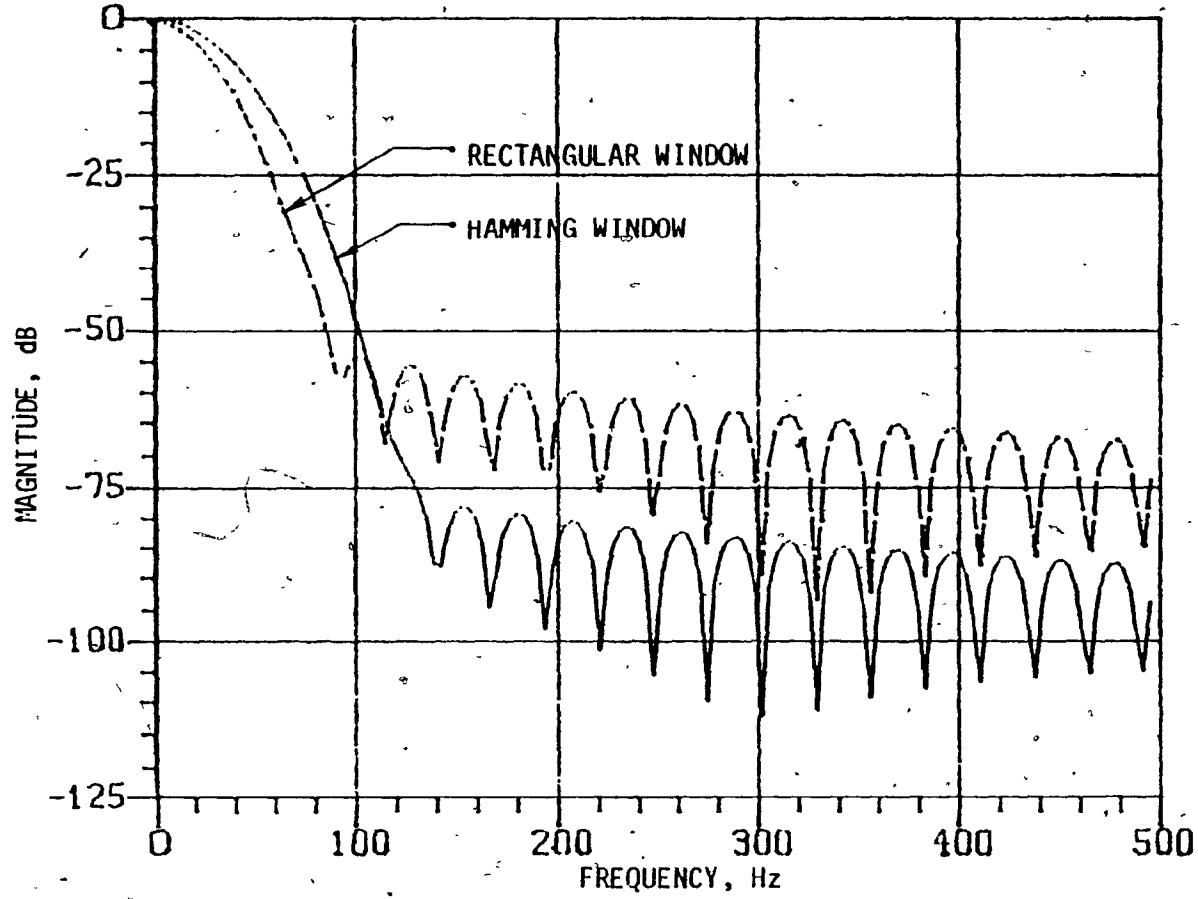


FIGURE 3.19: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ_c = 0 deg.

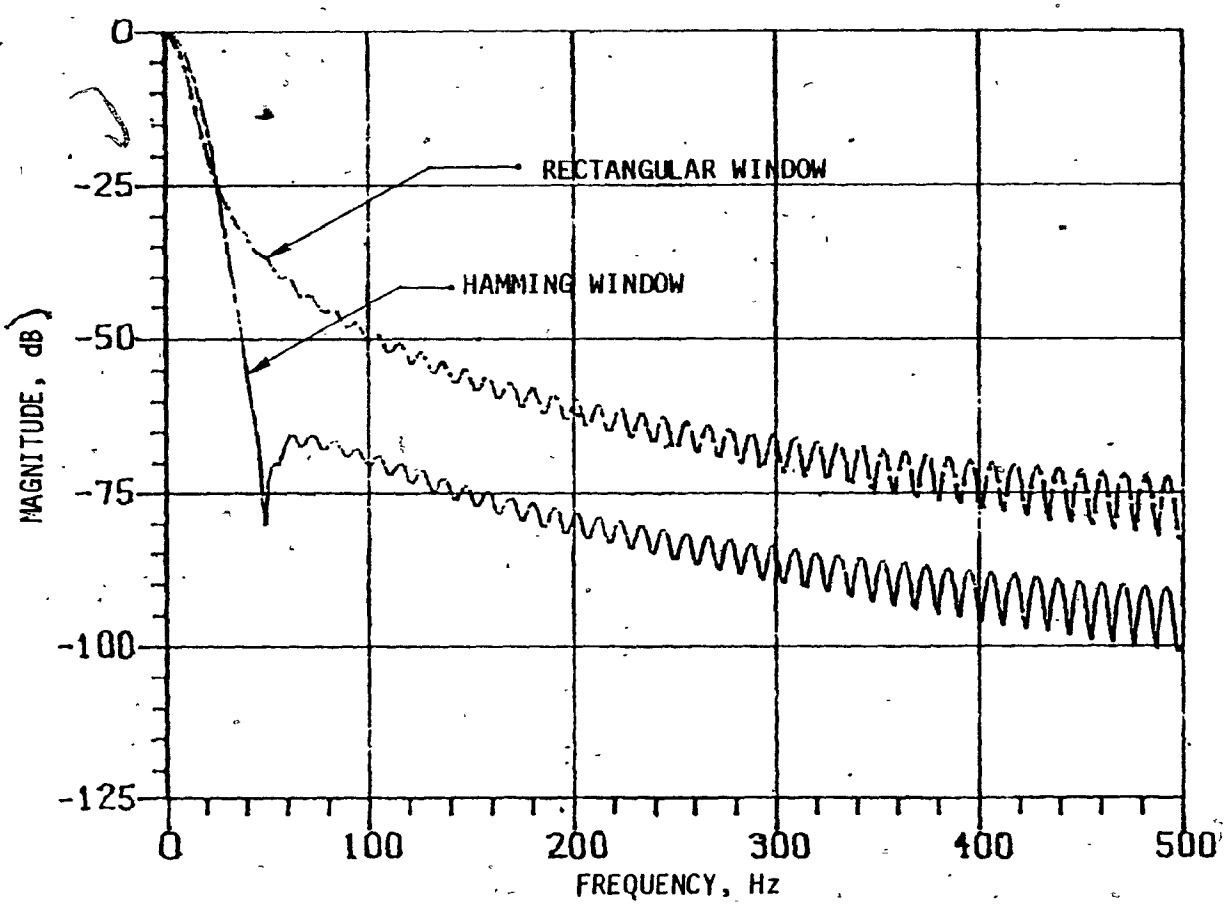


FIGURE 3.20: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 15 deg.

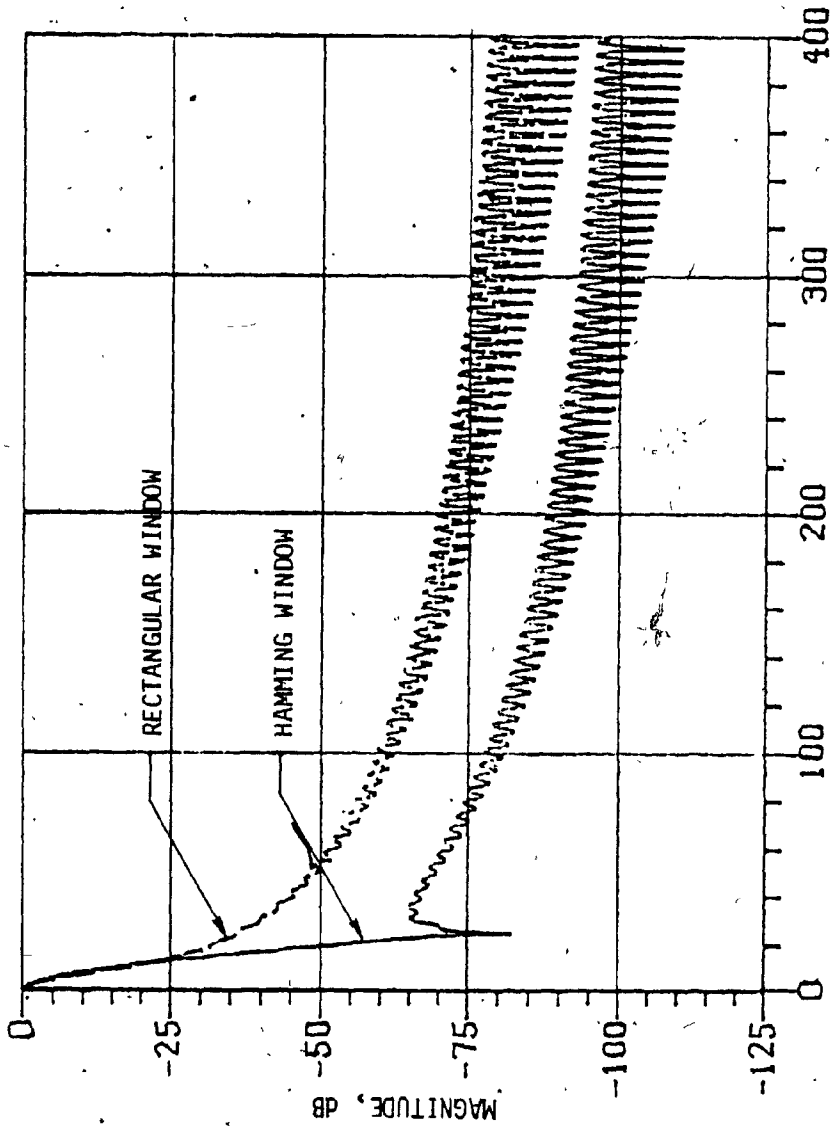


FIGURE 3.21: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 30 deg.

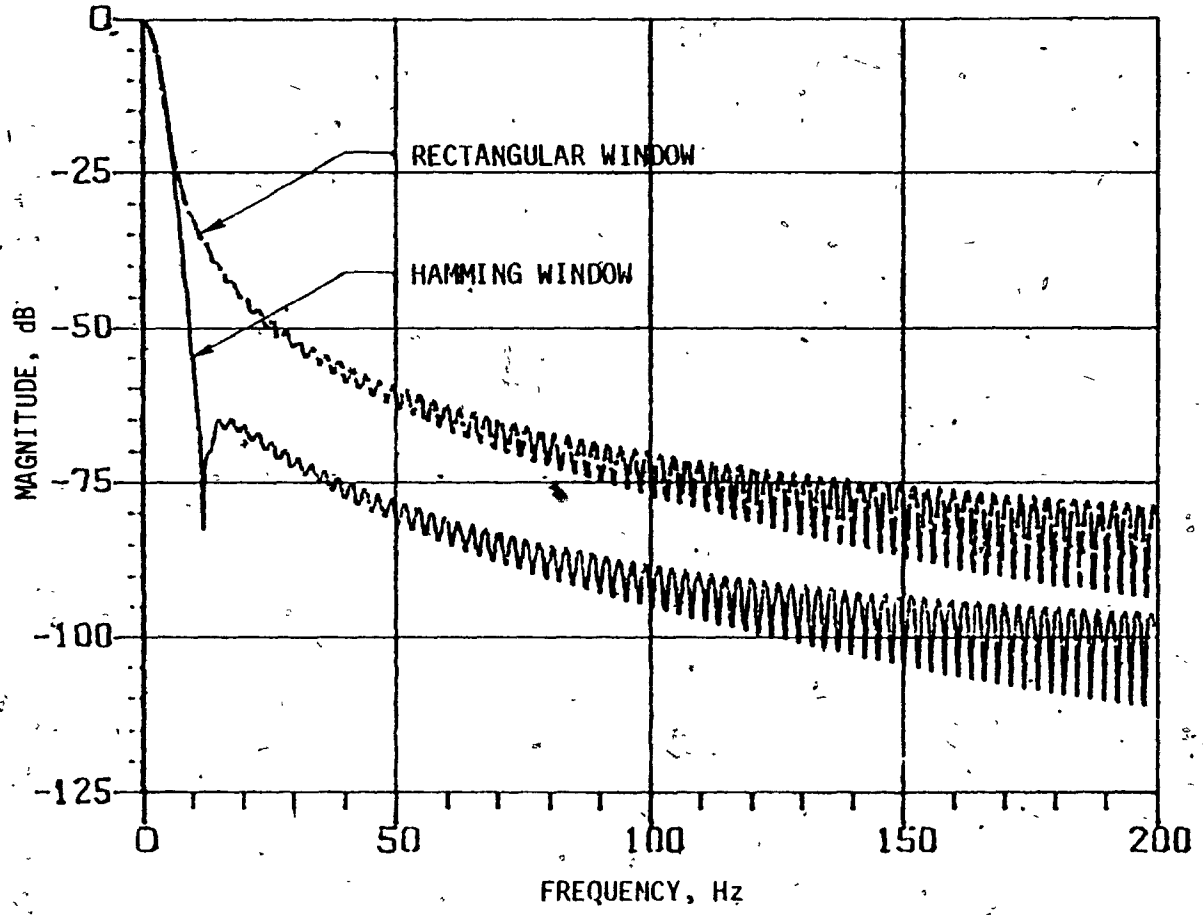


FIGURE 3.22: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 45 deg.

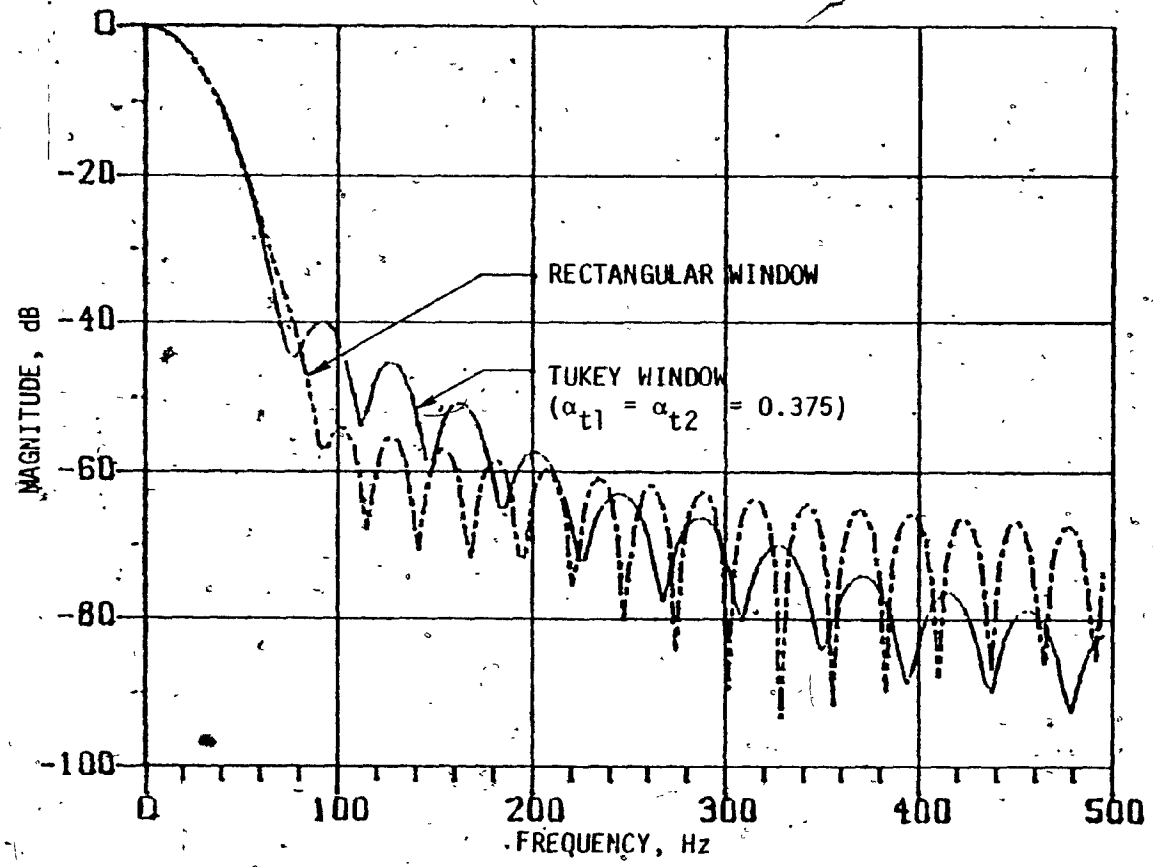


FIGURE 3.23: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 0 deg.

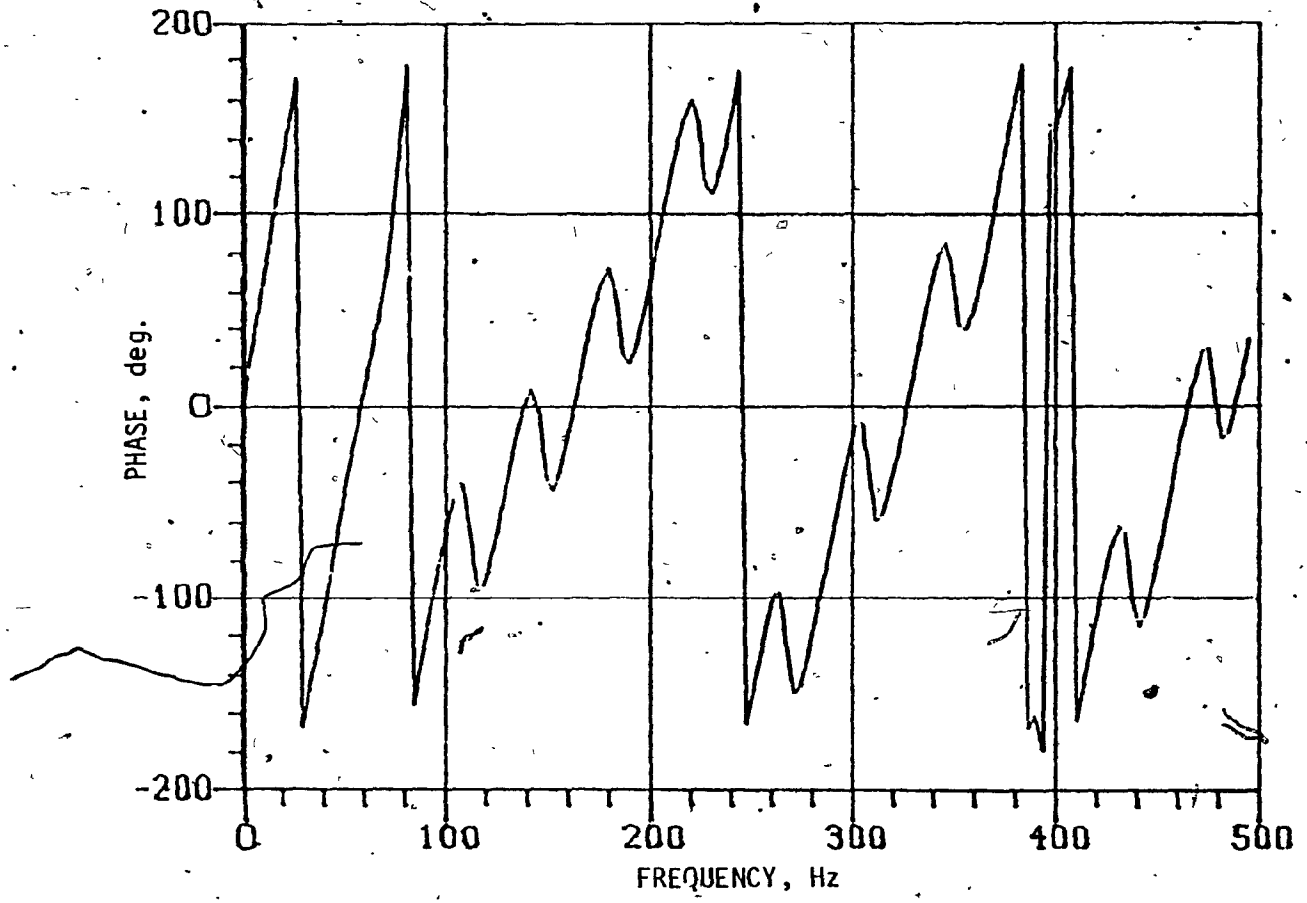


FIGURE 3.24: PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (TUKEY WEIGHTED) FOR Φ = 0 deg. and α_t = 0.375.

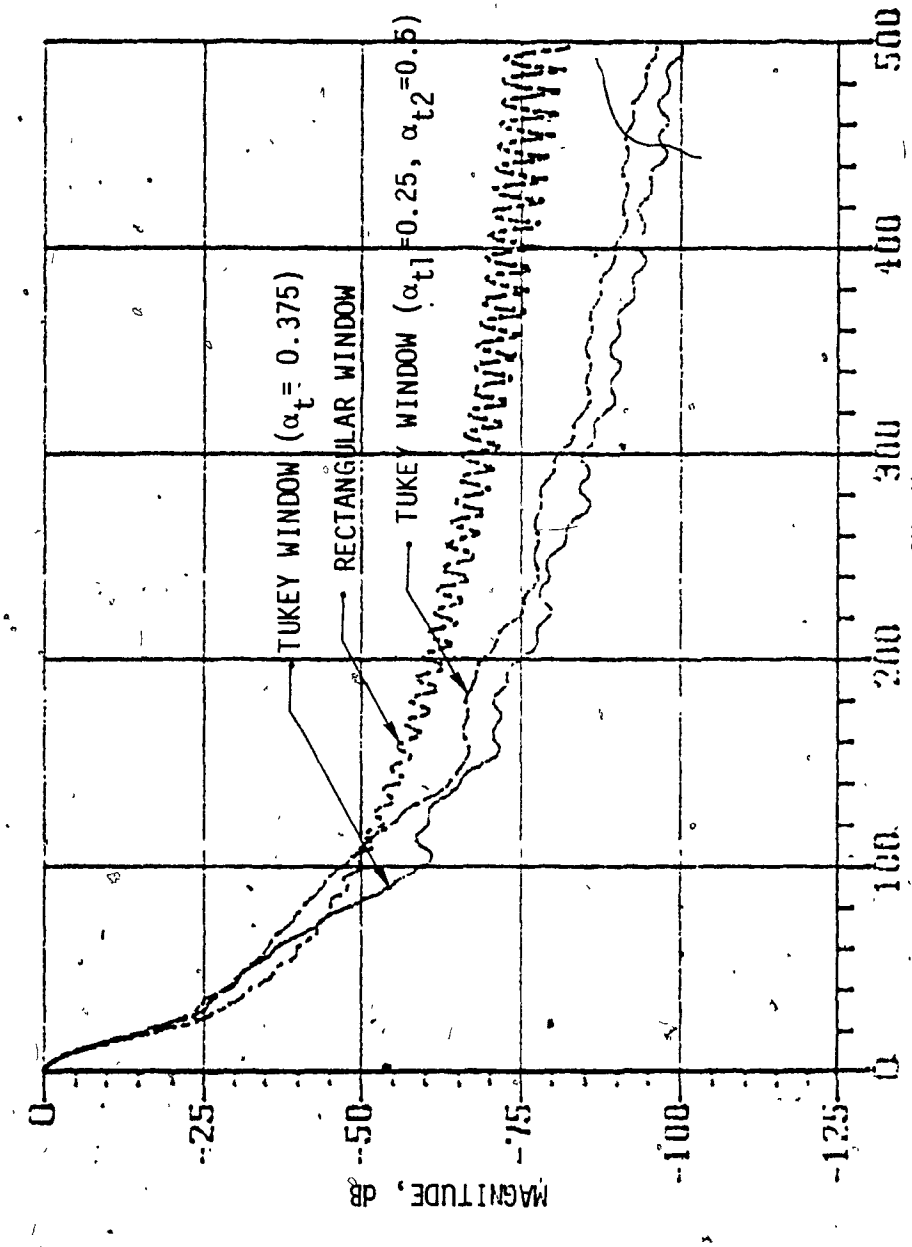


FIGURE 3.25: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR ϕ = 15 deg.

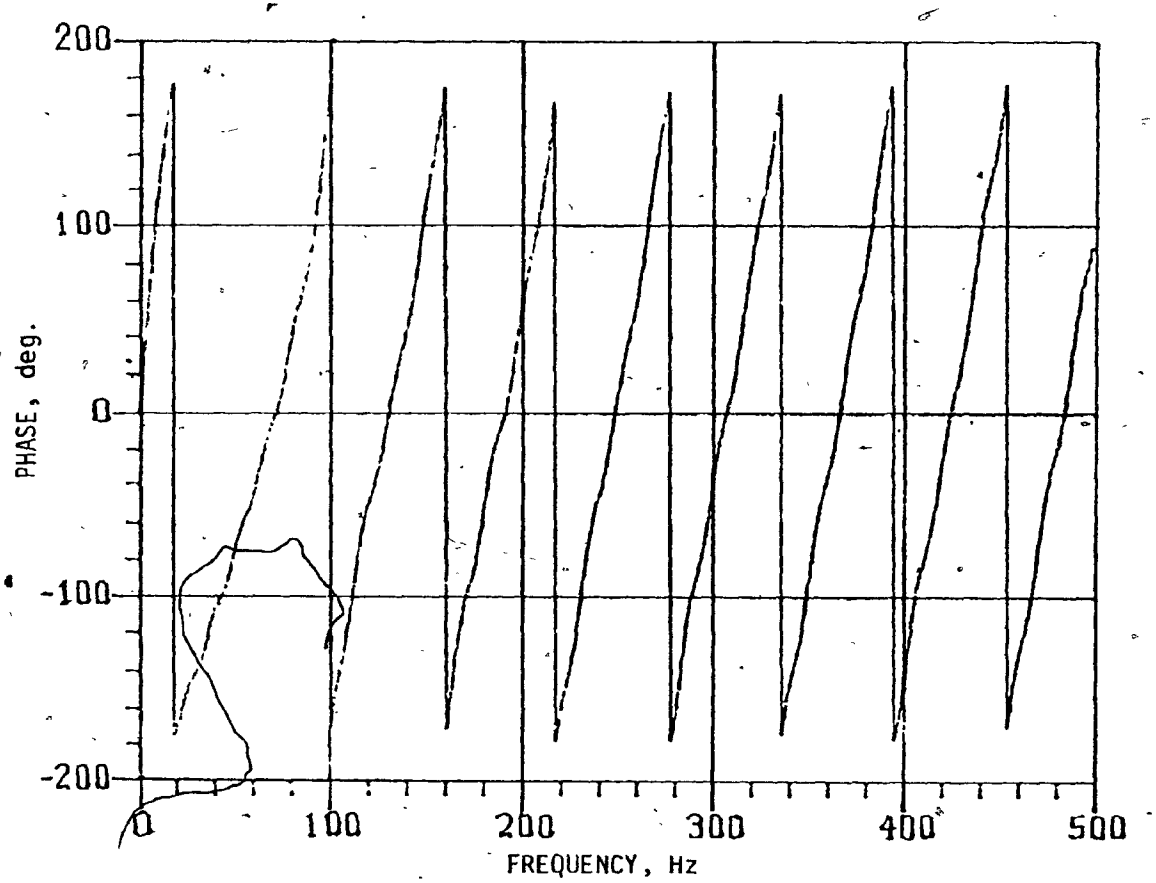


FIGURE 3.26: PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (TUKEY WEIGHTED)

FOR Φ = 15 deg. and α_t = 0.375.

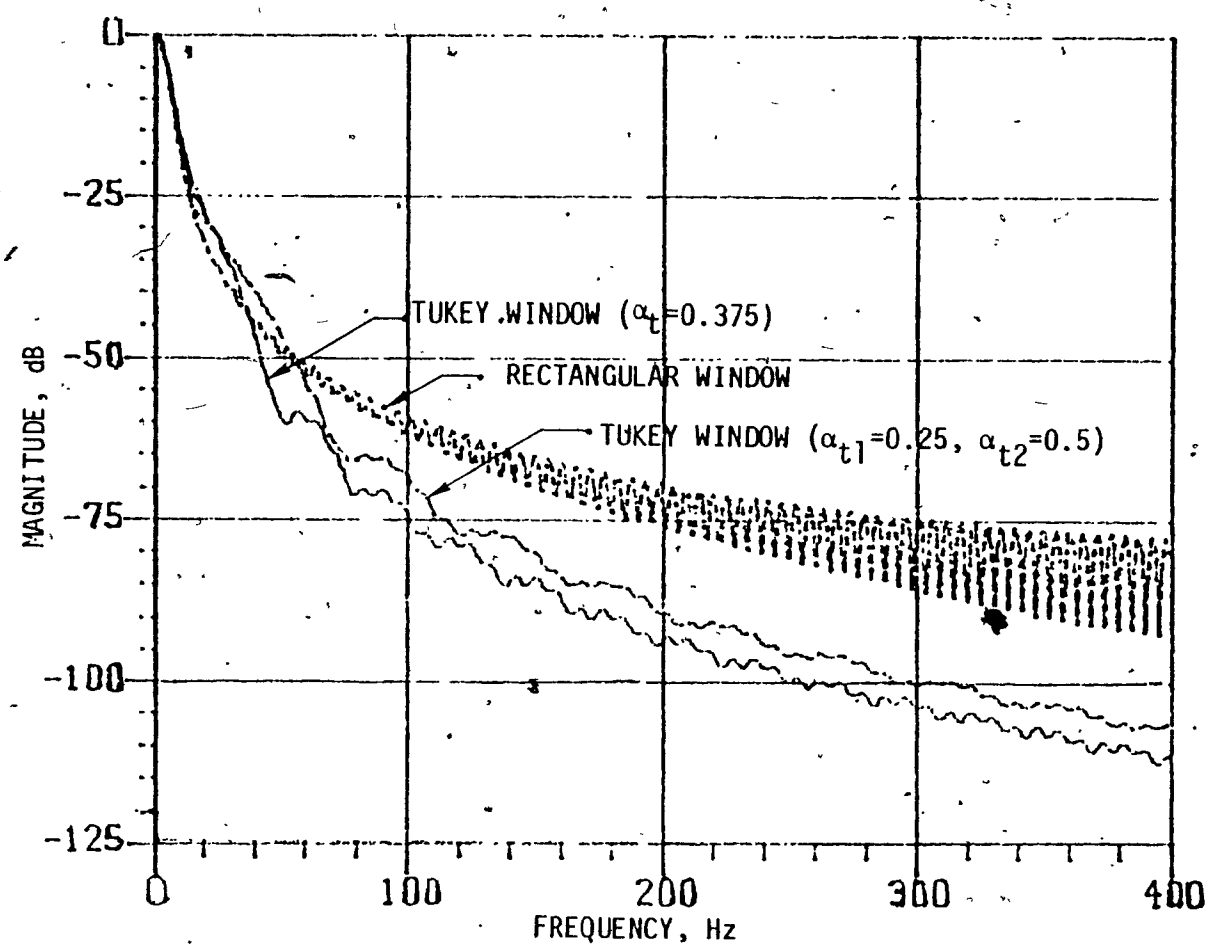


FIGURE 3.27: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR Φ = 30 deg.

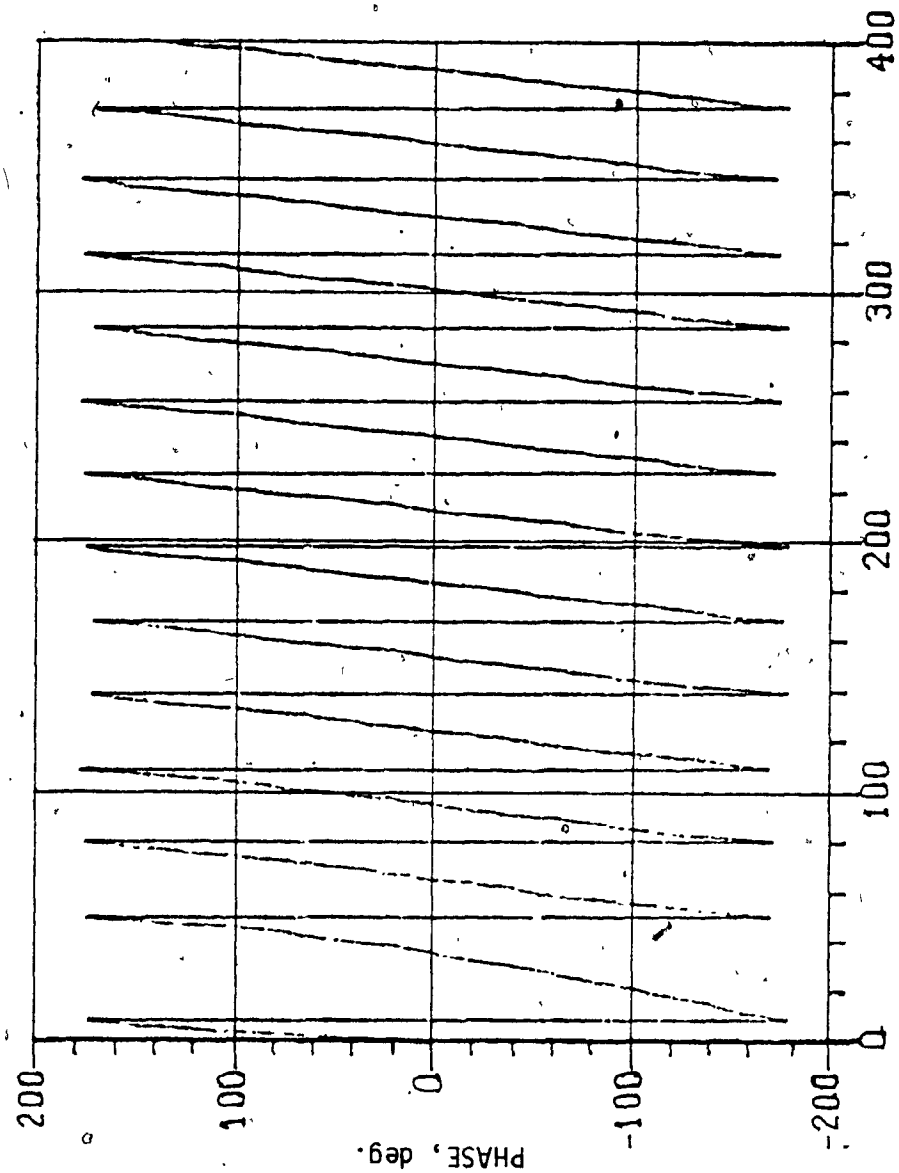


FIGURE 3.28: PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (TUKEY WEIGHTED) FOR Φ = 30 deg. and α_t = 0.375.

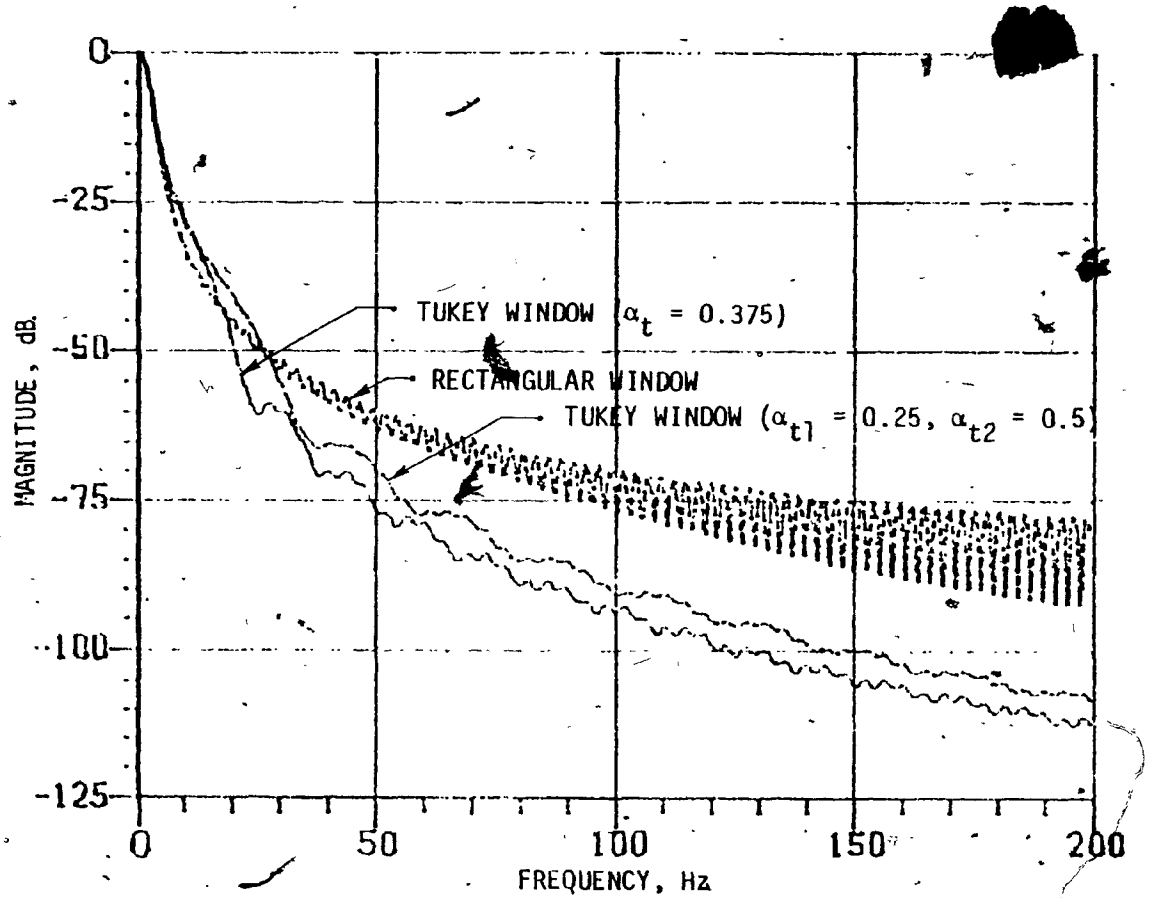


FIGURE 3.29: MAGNITUDE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (WEIGHTED) FOR ϕ = 45 deg.

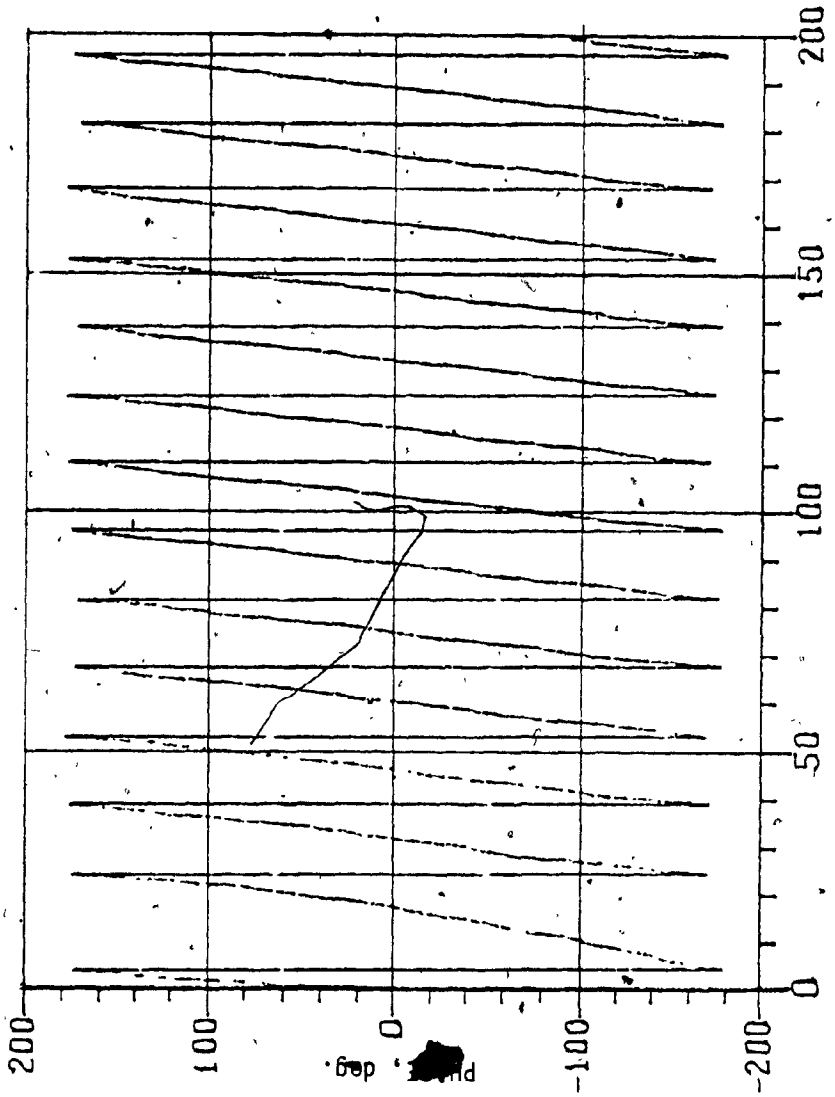


FIGURE 3.30: PHASE SPECTRUM OF BOTTOM RETURN SIGNAL ENVELOPE (TUKEY WEIGHTED)

FOR Φ = 45 deg. and α_t = 0.375.

proportion of the received signal envelope/data to be tapered is kept adjustable, a family of windows is obtained with the corresponding tapering at the ends and leaves the main portion at the center of the received signal envelope unweighted. So by using the Tukey window, the main portion of the received signal envelope sequence is kept unchanged, thereby achieving a close match of the -3 dB width to that of the expected echo duration at the cost of higher sidelobes. Also considering the practical point of view, the cosine, Hann and Hamming windows applied to the received signal envelope sequence of finite length K = 256 need 256 real multiplies of the time sequence, whereas Tukey window with $\alpha_t = \alpha_{t1} = \alpha_{t2} = 0.375$ needs only 48 real multiplies providing considerable hardware saving and high speed. But using the cosine and Hann windows as frequency windows, given by equations (3.37) and (3.39) respectively, the windowing is performed as 256 real adds and 256 binary shifts for the cosine window, 512 real adds and 512 binary shifts for the Hann window on the spectral data. (Note the loss of binary weighting for the Hamming window and hence the need to perform multiplication to apply the weighting factors of the spectral convolution). But since the windowing is pre-performed with the windowed coefficients stored in memory and not in real time, the attractive property of eliminating multiplies when the windowing is performed as a spectral convolution using DFT is of no real merit. Hence the selection of the Tukey window is justified. Tables 3.1 - 3.4 give the window function characteristics and allow a performance comparison between the various windows in terms of the major parameters for each steering angle. For all the steering angles, Tukey window with $\alpha_t = 0.375$ is selected and for steering angles away from the

TABLE 3.1

A COMPARISON BETWEEN VARIOUS WINDOW FUNCTIONS IN TERMS OF ITS PARAMETERS FOR STEERING ANGLE Φ = 0 deg.

WINDOW FUNCTION NAME	TIME WINDOW FUNCTION	FREQ. WINDOW FFT WEIGHTING FUNCTION	EQUIVALENT NOISE BW	-3dB BW	HIGHEST # SIDE LOBE LEVEL (dB)	# -3dB TIME WIDTH (s)
RECTANGLE	1	V(m)	$1.00/T_W$	$0.89/T_W$	-54	0.0105
COSINE	$\sin(\frac{k\pi}{K})$	$0.5H(m)-0.5H(m+1)$	$1.23/T_W$	$1.20/T_W$	-35.5	0.0091
HANN	$0.5-0.5\cos(\frac{2k\pi}{K})$	$-0.25H(m-1)+0.5H(m)$ $-0.25H(m+1)$	$1.50/T_W$	$1.44/T_W$	-90	0.0083
HAMMING	$0.54-0.46\cos(\frac{2k\pi}{K})$	$-0.23H(m-1)+0.54H(m)$ $-0.23H(m+1)$	$1.36/T_W$	$1.30/T_W$	-79	0.0085
TUKEY (α_t=0.375)	1 $0.5[1-\cos(\frac{k\pi}{m})]$ $0.5[1-\cos\frac{(K-k)\pi}{m}]$		$1.16/T_W$	$1.08/T_W$	-40	0.0105

See footnote on page 90

TABLE 3.2

A COMPARISON BETWEEN VARIOUS WINDOW FUNCTIONS IN TERMS OF ITS PARAMETERS FOR STEERING ANGLE Φ = 15 deg.

WINDOW FUNCTION NAME	HIGHEST # SIDE LOBE LEVEL (dB)	-3dB # TIME WIDTH (s)
RECTANGLE	-40	0.031
COSINE	-29	0.0258
HANN	-90	0.0239
HAMMING	-70	0.0245
TUKEY ($\alpha_{t1}=\alpha_{t2}=0.375$)	-60	0.030
TUKEY ($\alpha_{t1}=0.25$, $\alpha_{t2}=0.5$)	-65	0.031

See footnote on page 90

TABLE 3.3

A COMPARISON BETWEEN VARIOUS WINDOW FUNCTIONS IN TERMS OF ITS PARAMETERS FOR STEERING ANGLE Φ = 30 deg.

WINDOW FUNCTION NAME	HIGHEST # SIDE LOBE LEVEL (dB)	-3dB # TIME WIDTH (s)
RECTANGLE	-42	0.0602
COSINE	-29	0.0510
HANN	-95	0.0473
HAMMING	-65	0.0486
TUKEY ($\alpha_{t1}=\alpha_{t2}=0.375$)	-58	0.058
TUKEY ($\alpha_{t1}=0.25$, $\alpha_{t2}=0.5$)	-65	0.0602

See footnote on page 90

TABLE 3.4

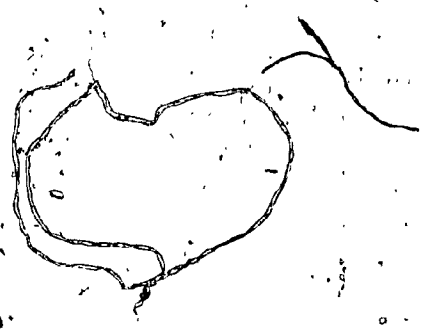
A COMPARISON BETWEEN VARIOUS WINDOW FUNCTIONS IN TERMS OF ITS PARAMETERS FOR STEERING ANGLE Φ = 45 deg.

WINDOW FUNCTION NAME	HIGHEST # SIDE LOBE LEVEL (dB)	-3dB # TIME WIDTH (s)
RECTANGLE	-42	0.123
COSINE	-29	0.104
HANN	-95	0.0969
HAMMING	-65	0.0999
TUKEY ($\alpha_{t1}=\alpha_{t2}=0.375$)	-58	0.118
TUKEY ($\alpha_{t1}0.25$, $\alpha_{t2}=0.5$)	-65	0.123

See footnote on page 90

vertical, Tukey window with $\alpha_{t1} = 0.25$ and $\alpha_{t2} = 0.5$ gives a comparable result. Thus the Tukey window, w(k) is applied as 48 real multiplies on the first 48 samples of the time sequence h(k) of the received signal envelope, the last 48 samples being identical, as the Tukey window for $\alpha_t = 0.375$ is symmetrical. The rest of the time sequence h(k) is unweighted. The windowed sequence $h_w(k)$ is retained as it is for further processing. It should be noted that if $h_w(k)$ is time reversed and shifted, it is the impulse response of the matched filter. The Dolph-Chebyshev window and the Kaiser-Bessel window [27] minimize the mainlobe width for a given sidelobe level. These window functions can give results comparable to those for the Tukey window. However, the Tukey window is much simpler and more economical and hence is selected for this application.

In Tables 3.1-3.4, the highest sidelobe level and the -3 dB time width given are the sidelobe levels of the spectrum of the weighted smooth bottom function and the -3 dB time width of the weighted smooth bottom function, respectively.



CHAPTER IV

CHAPTER IV

MATCHED-FILTER IMPLEMENTATION

4.1 BACKGROUND

Matched-filter implementation using high-speed convolution is considered. The high-speed convolution is simulated using DFT. Throughout this thesis DFT is computed using an efficient fast Fourier transform (FFT) algorithm [33]. The computation of the convolution using special time domain transforms replacing the DFT is considered for comparison. Since the transmitted signal is time limited, the operation of matched filtering is equivalent to filtering with non-recursive or finite impulse response (FIR) filter [20]. The received signal envelope, time-reversed, delayed and windowed, is the impulse response of the FIR matched filter.

4.2 HIGH SPEED CONVOLUTION USING FFT

High-speed convolution [32] is an efficient method of implementation for discrete-time FIR filters. Here the output of the matched filter given in equations (3.10) and (3.11) as a linear convolution in the time-domain is implemented in the frequency domain as a product, using the properties of DFT and is shown in Figure 4.1. The efficiency is achieved from using the fast Fourier transform (FFT) algorithm to compute the DFT's involved. The sequences b(k) and m(k) of length 256 are transformed using DFT, the transforms B(m) and M(m) are multiplied point by point, and the product U(m) is inverse transformed to give the output of the filter. These three basic steps may be performed using the radix-2 FFT. Then the output of the FIR matched filter is given by

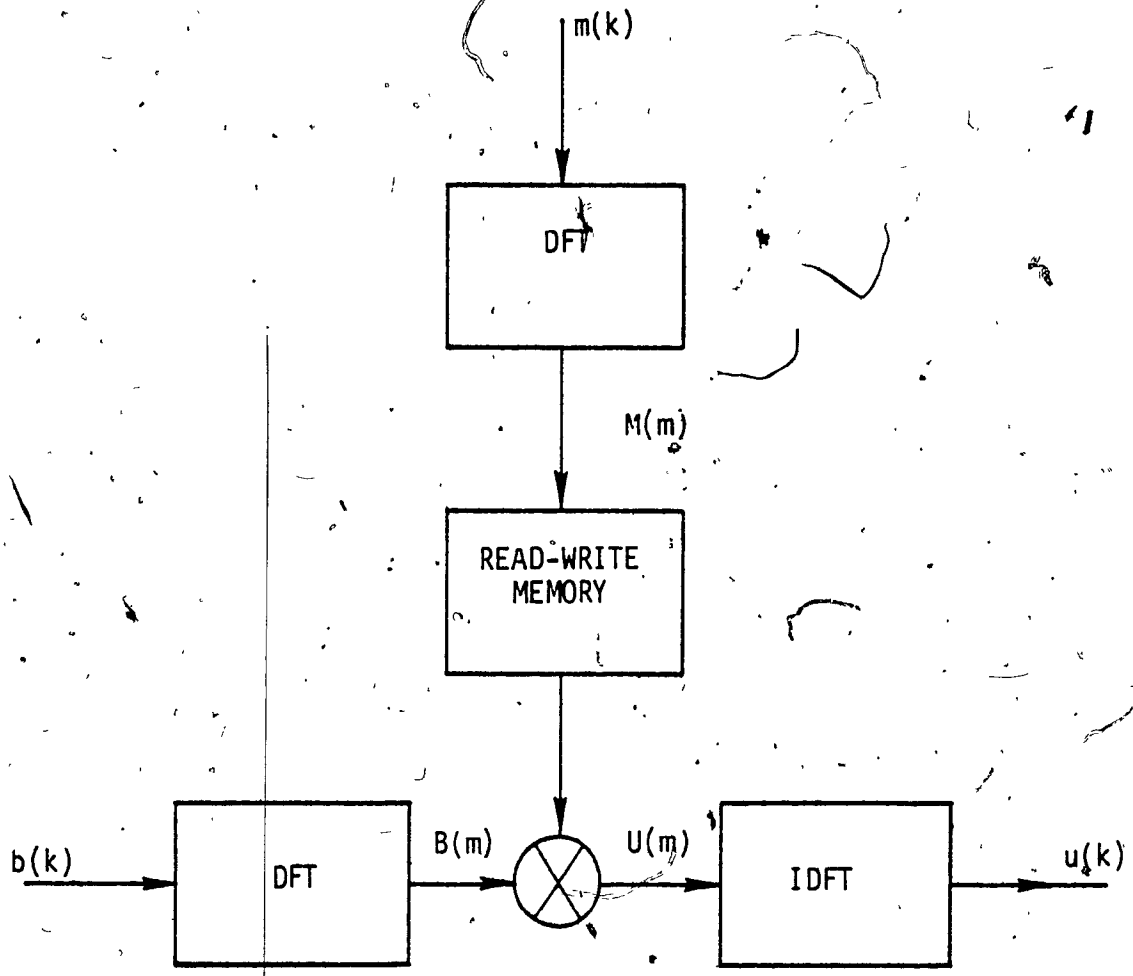


FIGURE 4.1: High Speed Convolution Using DFT

$$U(m) = B(m) \cdot M(m) \tag{4.1}$$

Now the equation (3.9) can be represented in the digital domain as

$$U(m) = B(m) \cdot H(-m) \quad e^{-j2\pi m T_w} \tag{4.2}$$

The exponential in equation (4.2) is a linear phase shift corresponding to a time delay required to make the matched filter realizable. Ignoring the time delay for the purpose of analysis, the output of the FIR matched filter can be given using equations (3.22) and (3.23) as

$$U(m) = B(m) \cdot H_w^*(m) \tag{4.3}$$

and is illustrated in figure 4.2.

The program CONVOL given in Appendix C simulates the high-speed convolution to obtain the output of the matched filter and is used as a test program.

The implementation of high-speed convolution using FFT [20,25,34] requires the calculation of two FFT's and the multiplication of two N-point sequences, if the DFT of the filter is pre-computed and stored in a read-write memory. Even though $b(k)$ and $h_w(k)$ are of finite duration sequences of length K, where K = 256, the output sequence of the filter $u(k)$ is obtained by the periodic convolution of the two sequences due to the implied periodicity in the DFT representation of finite duration sequences. Hence a cyclic convolution is performed which introduces the wrap-around error. In order to force the DFT to perform the desired linear convolution and eliminate the wrap-around error, both the sequences to be convolved must be appended with zeros to form new sequences of the same length N, where $N \geq 2K-1$. Therefore, the filter

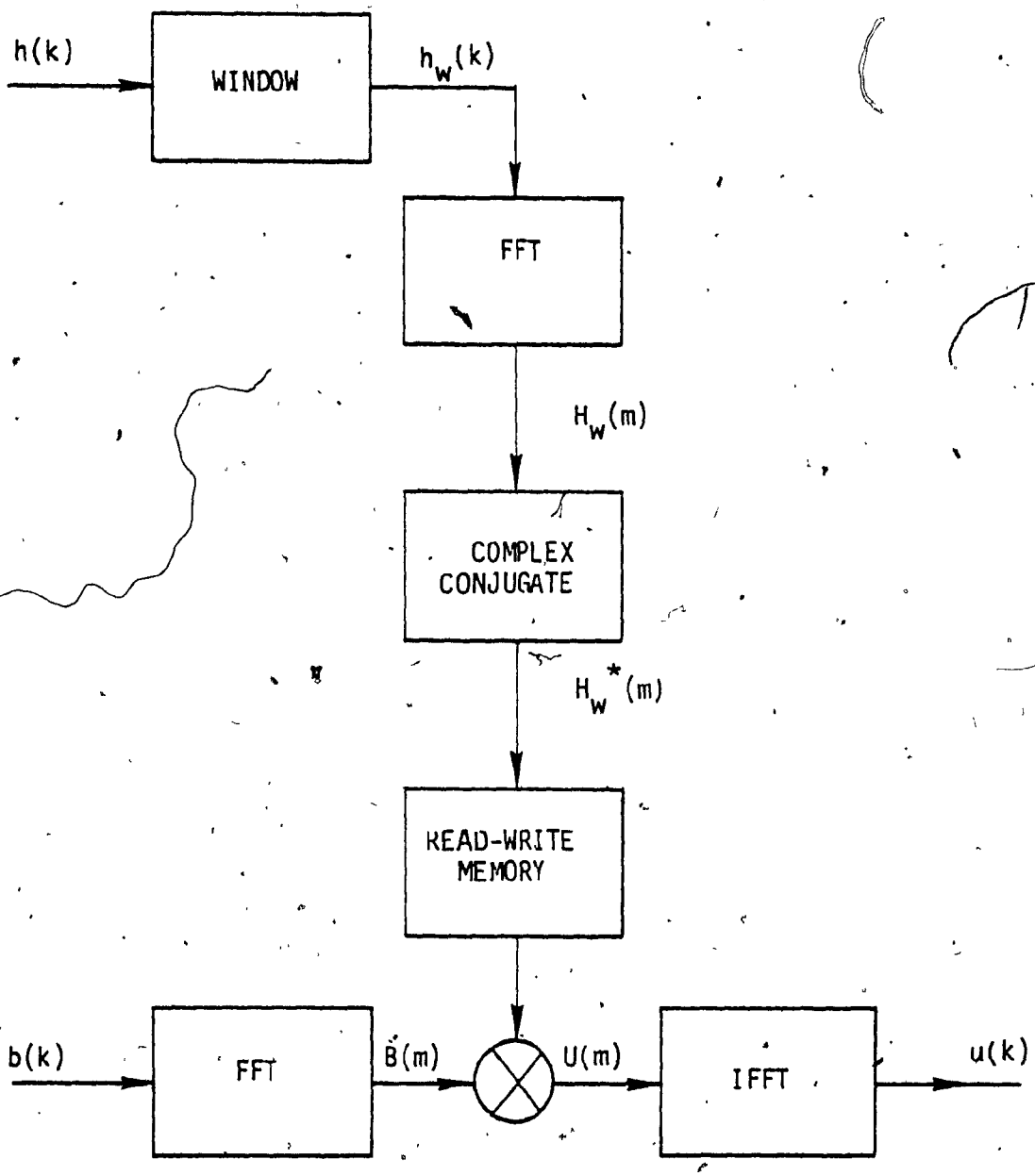


FIGURE 4.2: High Speed Implementation of Digital Matched Filter Using FFT

impulse response sequence $h_w(k)$ and the data sequence b(k) are padded with zeros to obtain the corresponding sequences of length N, where N is chosen as 512 (power of 2). Now the cyclic convolution that is obtained yields the same result as that of the desired linear convolution. The received data sequence b(k) is usually of very long duration. The convolution of this long sequence with a short finite impulse response sequence $h_w(k)$ by using the basic steps of high speed convolution involves a large DFT transform and is generally too large to compute. Also, there will be a large delay in the processing since no filtered output points can be obtained until all the input points have been collected. To eliminate the above drawbacks, the longer data sequence b(k) is "sectioned" into pieces and each section is then convolved with the finite impulse response sequence using the high-speed convolution as before and the filtered sections are fitted together in the appropriate way. Such a technique of "sectioning" or "block filtering" reduces the required amounts of computation, delay in the processing and memory. Two such methods are overlap-add and overlap-save.

4.2.1 Overlap-Save Method [25,35]

This method is used for the implementation purpose. The program "LAPSAVE" (Appendix C) calculates the filter output using this technique. The overlap-save technique corresponds to implementing a circular convolution of $h_w(k)$ with $b_n(k)$, and saving only that part of the circular convolution that corresponds to a linear convolution. In particular, the circular convolution of the 256 point impulse response of the matched filter with a 512 point section, the first 256 points are incorrect,

while the remaining points are identical to the output of a linear convolution. The input sequence b(k) is sectioned into lengths of 512 so that each input section overlaps the preceding section by 256 points and is given by

$$b_n(k) = b[k+n(256)] \quad , \begin{cases} 1 \leq k \leq 512 \\ n = 0,1,2 \end{cases} \qquad (4.4)$$

where n is the number of sections = 3 for input sequence length of 1024. Now each of these sections is circularly convolved with $h_w(k)$ to give sectioned output $s_n(k)$, i.e., computing the inverse DFT of the product of the DFT's of 512 point input sequence b(k) and the sequence $h_w(k)$ augmented with 256 zeros so that its sequence contains 512 samples. (It should be noted that the sequence must be augmented so that the new length is at least twice that of the original length). The portion of each output section in the range $1 \leq k \leq 255$ is obtained due to wrap-around error and is discarded, and the remaining points of each output section are joined to construct the final output and is given as

$$u(k) = \sum_{n=0}^{2} u_n[k+255-n(256)] \quad , \; 1 \leq k \leq 768 \qquad (4.5)$$

Since each input section consists of first 256 points, saved from the last 256 points of the previous section and the remaining last 256 points being new, each output section contains only the last 256 points identical to those that would be obtained by a linear convolution, while the remaining points are incorrect and are being discarded.

Using the high-speed convolution, the reduction in computation proportional to $\log_2 N/N$ operations to that of direct convolution can be achieved. Further reduction in multiplication can be obtained by making use of the symmetry property of DFT for real data sequences. A 512 point DFT on the complex sequence $c_{n1}(k)$ is computed where $c_{n1}(k)$ is defined as

$$c_{n1}(k) = b_{n1}(k) + j\, b_{n2}(k) \tag{4.6}$$

where $b_{n1}(k)$ and $b_{n2}(k)$ are the sectioned input sequence for steering angles $\pm\, \Phi$ deg.

Then their transforms can be obtained as

$$\begin{aligned} B_{n1}(m) &= [\, C_{n1}(m) + C_{n1}^{*}\,(512-m)\,] \,/\, 2 \\ B_{n2}(m) &= [\, C_{n1}(m) - C_{n1}^{*}\,(512-m)\,] \,/ 2j \end{aligned} \tag{4.7}$$

This requires extra 512 additions but approximately half the multiplications are saved over the straightforward transforming of each sequence separately. The disadvantage of FFT implementation is the introduction of roundoff error due to finite register length, the algorithm, the radix and the arithmetic. Figure 4.3 gives the schematic diagram of the FFT implementation of the matched filter.

4.3 CONVOLUTION USING SPECIAL TRANSFORMS

While the direct calculation of the convolution would require a number of multiplications and additions proportional to N^2 for large N, the use of FFT reduces this proportional to $N \log_2 N$ operations. Recently Agarwal and Burrus showed that formulating a one-dimensional convolution as a multidimensional convolution of the original signals [36] and using various fast algorithms for the short length convolutions results in

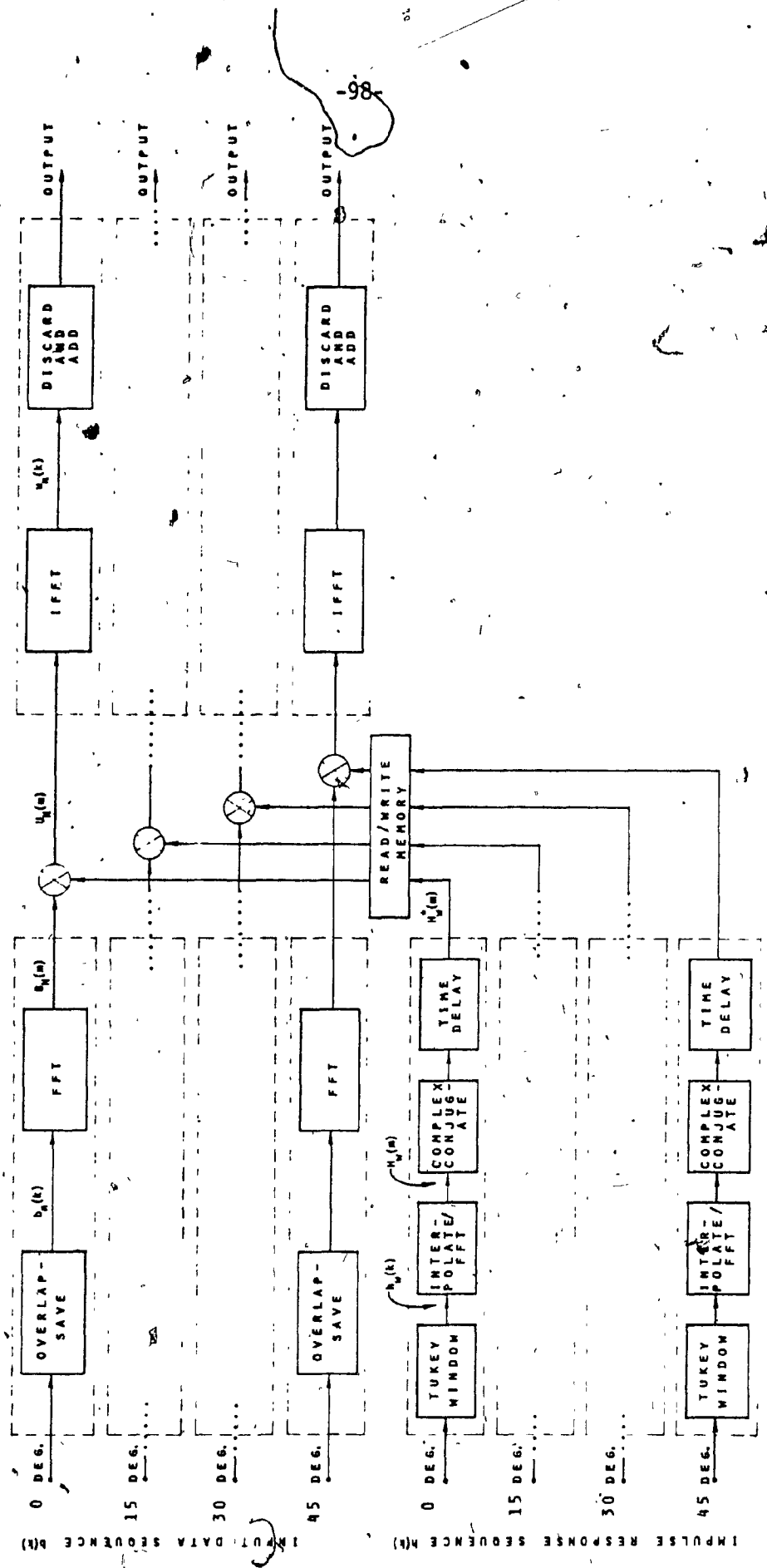


FIGURE 4.3: FFT IMPLEMENTATION OF MATCHED FILTER

improved multiplication efficiency for sequence lengths up to around 128. This is achieved only at the cost of increase in additions and the computational complexity. They also investigated transforms (number theoretic transforms) having cyclic convolution properties [37] and of the form

$$X(n) = \sum_{k=0}^{N-1} x(k)\, \alpha^{nk} \qquad n = 0,1,\ldots,N-1 \qquad (4.8)$$

where α is a root of unity of order N. Any transform having this cyclic convolution property also has a fast computational algorithm. For DFT,

$$\alpha = e^{-j2\pi/N}$$

is a primitive Nth root of unity in the complex number field. The complex number field can be replaced by a finite ring of integers under modulo F; and an integer α of order N replacing $e^{-j2\pi/N}$ used in a DFT. In this ring, when two integer sequences b(k) and m(k) are convolved, the output integer sequence u(k) is congruent to the convolution of b(k) and m(k) modulo F. These number theoretic transforms seem to be more attractive and computationally efficient in comparison to FFT for implementing convolution.

Agarwal and Cooley [38] showed that Chinese Remainder Theorem [39] can be used to map one-dimensional cyclic convolution to a multidimensional cyclic convolution which is cyclic in all dimensions and avoiding the necessity of appending zeros in order to use the cyclic convolution algorithms. Then special algorithms can be used to compute the short convolutions in each of the dimensions. The general form of the algorithm is given as

$$\underline{u} = C[(A\underline{m}) \times (B\underline{b})] \qquad (4.9)$$

where $\underline{m}$ and $\underline{b}$ are N element column vectors, and x denotes element-by-element multiplication. The rectangular matrices A and B are transforms of $\underline{m}$ and $\underline{b}$ respectively, and C transforms the product back to data space. Now the one-dimensional can be implemented as a multidimensional cyclic convolution with rectangular transforms by factoring N in to more than two mutually prime factors.

For $N = 504$,

$$N = r_1 \cdot r_2 \cdot r_3 = 7 \cdot 8 \cdot 9 \tag{4.10}$$

where r_1, r_2 and r_3 are mutually prime. The multidimensional index mapping is defined by

$$i_j \equiv i \bmod r_j \tag{4.10}$$

$$0 \leq i_j < r_j \quad , \quad j = 1,2,3$$

and the inverse mapping is

$$i \equiv i_1 u_1 + i_2 u_2 + i_3 u_3 \bmod N \tag{4.12}$$

where $u_j = q_j(N/r_j)$ in which q_j satisfies

$$q_j(N/r_j) \equiv 1 \bmod r_j \tag{4.13}$$

With this three-dimensional mapping, the one-dimensional cyclic convolution

$$u(k) = \sum_{n=0}^{N-1} b(n)\, m(k-n) \tag{4.14}$$

can be written as a three-dimensional cyclic convolution of the form

$$u(k_1,k_2,k_3) = \sum_{n_3=0}^{r_3-1} \sum_{n_2=0}^{r_2-1} \sum_{n_1=0}^{r_1-1} b(n_1,n_2,n_3)\, m(k_1-n_1,k_2-n_2,k_3-n_3) \tag{4.15}$$

The three-dimensional transformation is a direct product of three one-dimensional transformations having the cyclic convolution property of length r_1, r_2 and r_3, respectively, and can be written as

$$\underline{u} = C_1C_2C_3 [(A_3A_2A_1 \underline{m}) \times (B_3B_2B_1 \underline{b})] \quad (4.16)$$

The $B_3B_2B_1\underline{b}$ denotes a three-dimensional rectangular transform of $\underline{b}$. First the r_1-point transform $B_1\underline{b}$ is computed with respect to the first index n_1 of $\underline{b}$ for fixed values of all other indices, then compute the r_2-point transform with respect to n_2 for each fixed set of values of all other indices and then the r_3-point transform with respect to n_3 for each fixed value of the other indices. The inverse operation with c_j's is to be performed in a similar fashion as mentioned before but in the reverse order.

If A_j and M_j are the number of additions and multiplications, respectively, required for a length r_j one-dimensional convolution, then the total number of multiplications required for the three-dimensional cyclic condition is

$$M(r_1,r_2,r_3) = M_1M_2M_3 = 5852$$

and the total number of additions required is

$$A(r_1,r_2,r_3) = A_1r_2r_3 + M_1A_2r_3 + M_1M_2A_3 + M_1M_2M_3 = 34,678$$

To minimize the number of additions, the ordering of the indices r_1,r_2,r_3 should be according to the size of

$$T(r_j) = (M_j - r_j)/A_j \quad (4.17)$$

such that $T(r_k) \leq T(r_j)$ when $k < j$

So for the 504 point transform, the ordering of indices are 8,9,7.

The minimum number of multiplications required for computing a convolution is given by Winograd [40] and is 2N-L where L is the number of integers that divide N, including 1 and N.

The Table 4.1 gives the number of multiplications and additions for the computation of convolution by rectangular transform and FFT methods. It is seen that the rectangular transform technique of computing convolution has improved multiplication efficiency for sequence lengths up to around 420 [38]. For convolution of real sequences, these techniques need only real arithmetic as compared with complex arithmetic required by the FFT algorithm. This reduces hardware complexity considerably. Since the length of the cyclic convolution must be $N \geq 512$, for filter length of 256, the FFT implementation with optimum length # $N = 2048$ gives the best multiplication efficiency.

Agarwal and Burrus [41] suggested Fermat number transforms (FNT) for the computation of cyclic convolution in the longest dimension. This reduces the multiplications considerably. The only multiplications

---

The optimum length is chosen such that

$$F_2(K,N) = F_1(N)\, N/(N-K+1) \quad \text{is minimum}$$

where $F_2(K,N)$ is the number of multiplications per output point

F_1 is the number of multiplications per point required for a length N cyclic convolution

and K is the filter length.

TABLE 4.1

A COMPARISON BETWEEN FFT AND RECTANGULAR TRANSFORM METHODS

METHOD	FILTER LENGTH K	LENGTH OF THE CYCLIC CONVOLUTION N	OPTIMUM LENGTH N	FACTORS (ORDERED)	NO.OF MULTIPLICATIONS M_j	$F_1(N)$	MULTIPLICATIONS PER OUTPUT POINT $F_2(K,N)$	WINOGRAD'S THEORETICAL MINIMUM
FFT	256	512	-	2^9	5,124	10.00	20.00	1014
	256	-	2048	2^{11}	26,628	13.00	14.85	4084
RECTANGULAR TRANSFORM	256	504	-	8,9,7	5,852	11.61	23.5	2160
	256	-	840	3,8,5,7	10,640	12.66	18.20	4608

required for an FNT implementation of cyclic convolution are the N multiplications required to multiply the transforms. The main disadvantage is the maximum transform length for an FNT is proportional to the wordlength of the machine used and also a special purpose machine must be available. Now the FNT can be used with the rectangular transform technique discussed before [38]. For N = 1920, the length of the first dimension is taken as 128 (the maximum transform length of FNT) and the lengths of the other dimensions are taken as mutually prime odd numbers. Thus

$$N = 128 \cdot 3 \cdot 5 \tag{4.18}$$

The rest of the procedure is the same as that of the rectangular transform technique. Table 4.2 gives a comparison of multiplication efficiency for FNT and FFT implementation. The multiplications are reduced by almost five times but only at the cost of extra adds and the availability of a special purpose machine for the computation of integer arithmetic. The schematic diagram for the implementation of convolution by these special transforms is the same as that of FFT implementation shown in Figure 4.3, with FFT and IFFT blocks replaced by the corresponding forward and inverse rectangular or number-theoretic transforms.

TABLE 4.2

A COMPARISON BETWEEN FFT AND FNT FOR MULTIPLICATION EFFICIENCY

METHOD	LENGTH OF THE CYCLIC CONVOLUTION N	FACTORS	NUMBER OF MULTIPLICATIONS $F_1(N)$	MULTIPLICATIONS PER OUTPUT POINT $F_2(K,N)$
FFT	2048	2^{11}	13.00	14.85
FNT	1920	128 x 3 x 5	2.66	3.07

CHAPTER V

CHAPTER V

PERFORMANCE EVALUATION AND COMMENTS

The envelope of the bottom return signal $h(\Phi,d)$ for the different beam steering angles is represented as in equation (2.29). Figures 2.9 through 2.13 represent the synthesized smooth bottom function for 0, 15, 30 and 45° steering angles, respectively, and Table 2.2 shows that the expected echo duration is closely matched to the -3 dB width of the bottom return signals obtained from the model. The model is very flexible and the various parameters that represent it can be controlled to obtain the desired characteristics of the signal. Using the smooth bottom function, the bottom return signal with additive noise and the amplitude fluctuations of the signal for each of the steering angles is simulated as shown in Figure 2.8. Figures 2.14 through 2.17 show the simulated input data record $b(k)$ of length 512 points, the equivalent of twice the -40 dB time bandwidth of the respective beam steering angle. Since the signal duration is of length 256 points, the additive noise is added on either side of the signal to create an input data length of 512 points. The time at which it peaks corresponds to the time of the arrival of the bottom return signal on the MRA of the beam. The signal processor input SNR referenced to 300 HZ additive noise bandwidth is kept at approximately + 15 dB for each steering angle.

The smooth bottom function is windowed using a Tukey window with tapering coefficients $\alpha_{t1} = \alpha_{t2} = 0.375$, for each of the beam steering angles. Tables 3.1-3.4 illustrate comparison of the various windows considered in terms of certain criteria. The Tukey window satisfies

the performance criteria specified for the window selection. The windowed sequence, $h_w(k)$ implicitly represents the finite impulse response of the bottom return matched filter.

The windowed sequence is then correlated with the input data sequence b(k) to give the matched filter output. The windowed sequence for each of the steering angles is shown in Figures 5.1-5.4. The input data sequence b(k) of length 768 points for each of the beam steering angles, Φ = 0, 15, 30 and 45°, at an operating depth of 12000 ft., is shown in Figures 5.5-5.8. For these figures, the ordinate axis is linear in amplitude and normalized to unity. The High Speed convolution using FFT with overlap-save technique is performed for each data sequence with the corresponding impulse response of the filter. The simulated bottom matched filter output for beam steering angles of 0, 15, 30 and 45°, is shown in Figures 5.9-5.12. For these figures the ordinate axis is linear in amplitude. The matched filter output results demonstrate the effectiveness of the filter in smoothing out the background noise while producing a signal return with a well defined peak value.

The output of the bottom return matched filter is fed to a peak detector which determines the highest amplitude point and corresponding time position in the filter output data record. The time integer k corresponding to the peak amplitude point at the filter ouput data record that is obtained is given as

$$k = k_1 - k_2 + K + \varepsilon \qquad (5.1)$$

where k_1 is the time integer corresponding to the peak amplitude point of the input data record (to be determined).

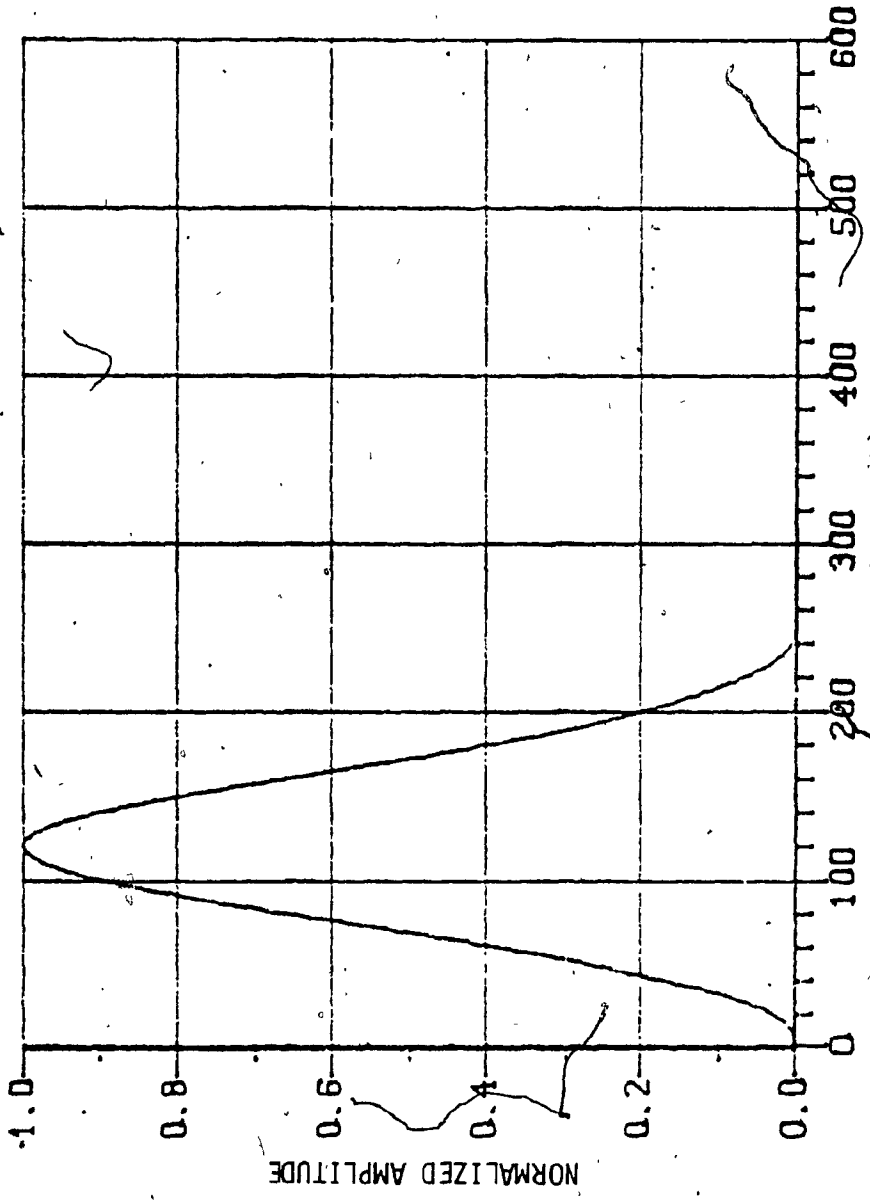


FIGURE 5.1: TUKEY WEIGHTED FINITE IMPULSE RESPONSE (Φ = 0 deg.)

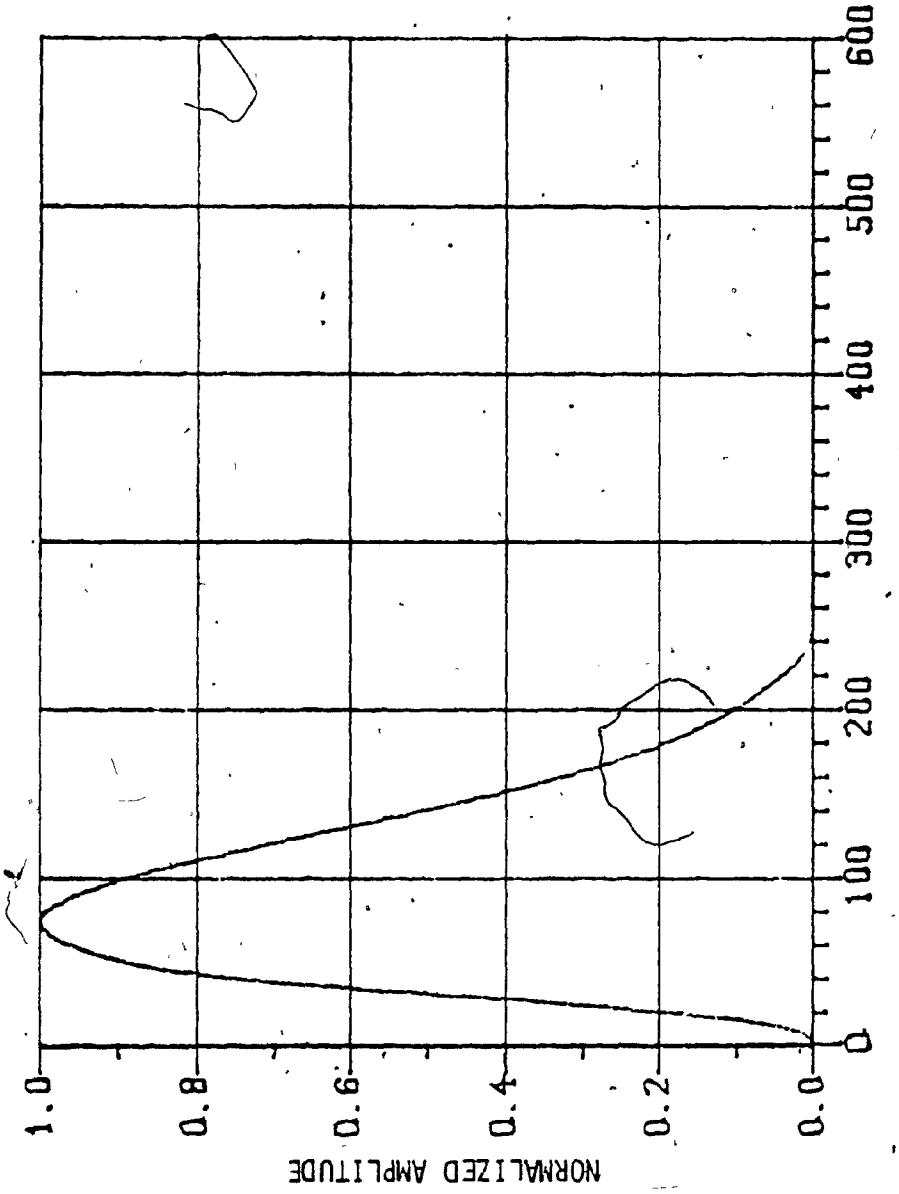


FIGURE 5.2: TUKEY WEIGHTED FINITE IMPULSE RESPONSE (ϕ = 15 deg.)

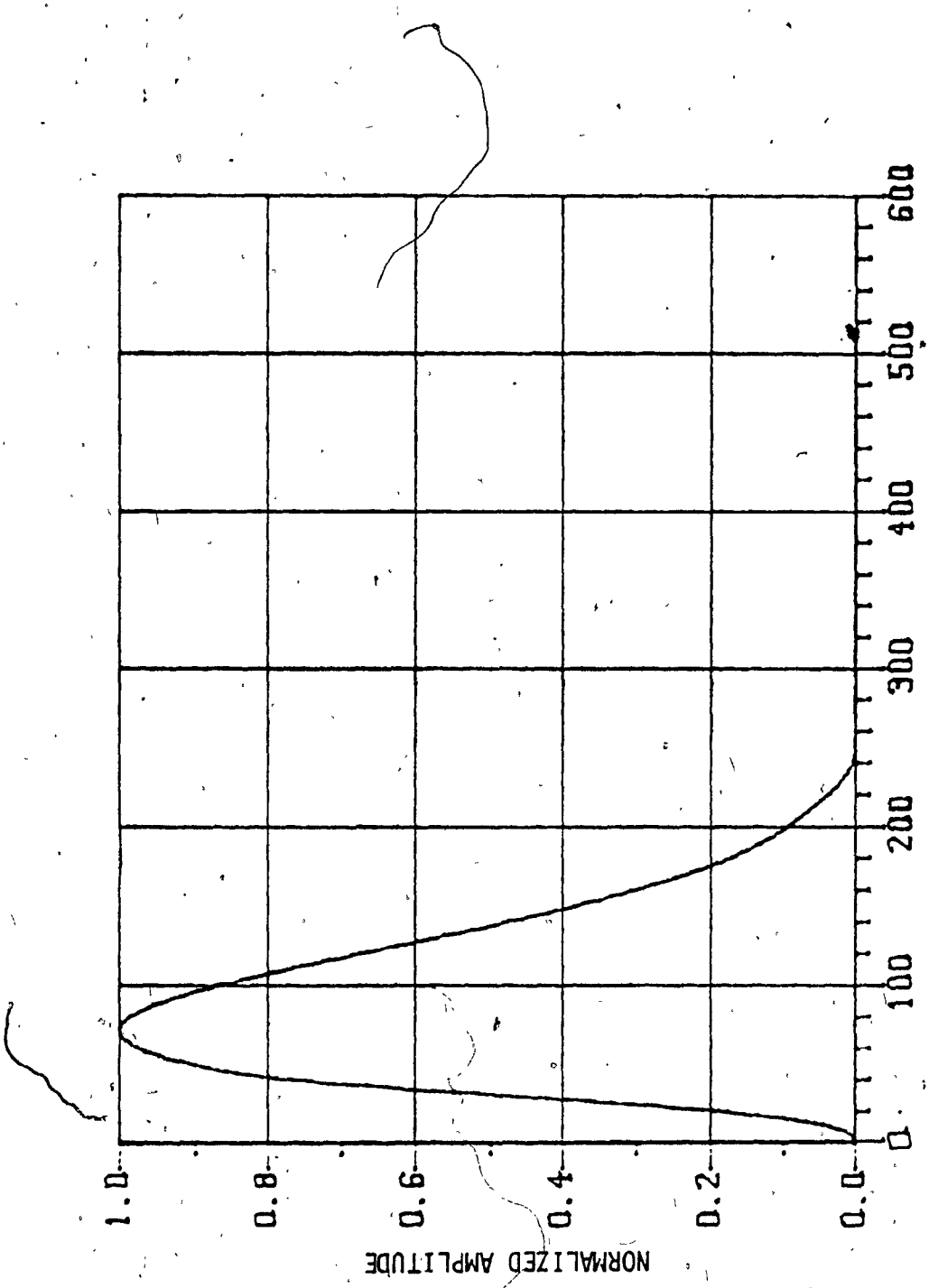


FIGURE 5.3: TUKEY WEIGHTED FINITE IMPULSE RESPONSE (ϕ = 30 deg.)

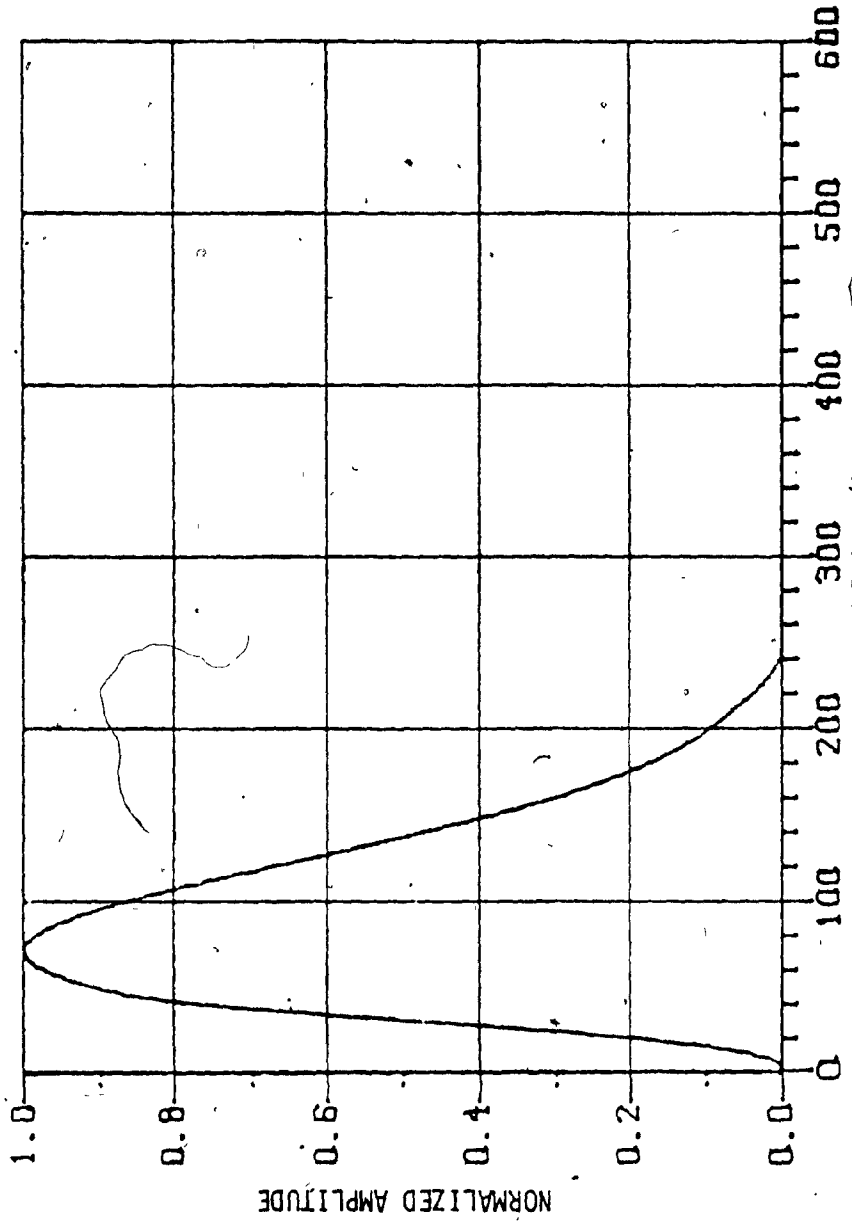


FIGURE 5.4: TUKEY WEIGHTED FINITE IMPULSE RESPONSE (Φ = 45 deg.)

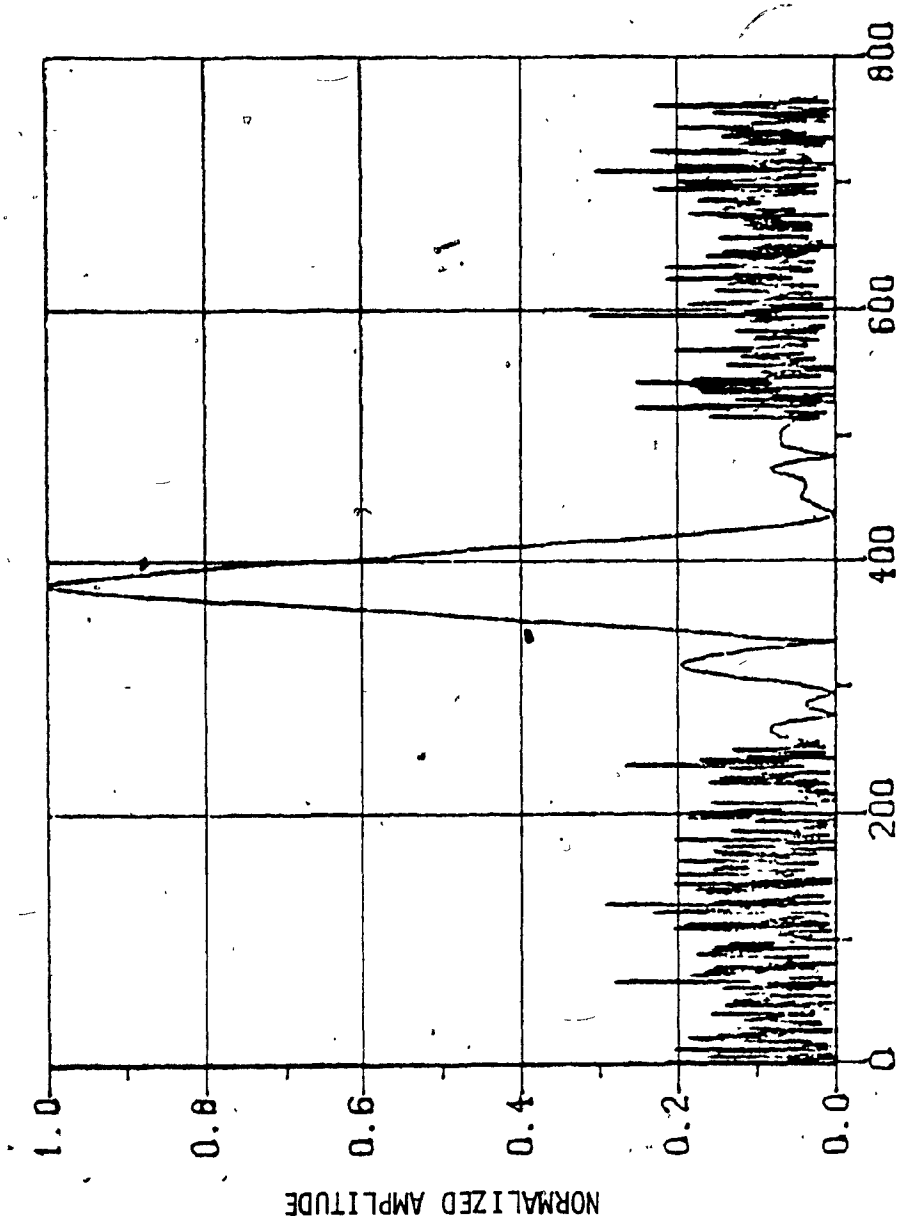


FIGURE 5.5: INPUT BOTTOM RETURN SIGNAL WITH NOISE FOR Φ = 0 deg.

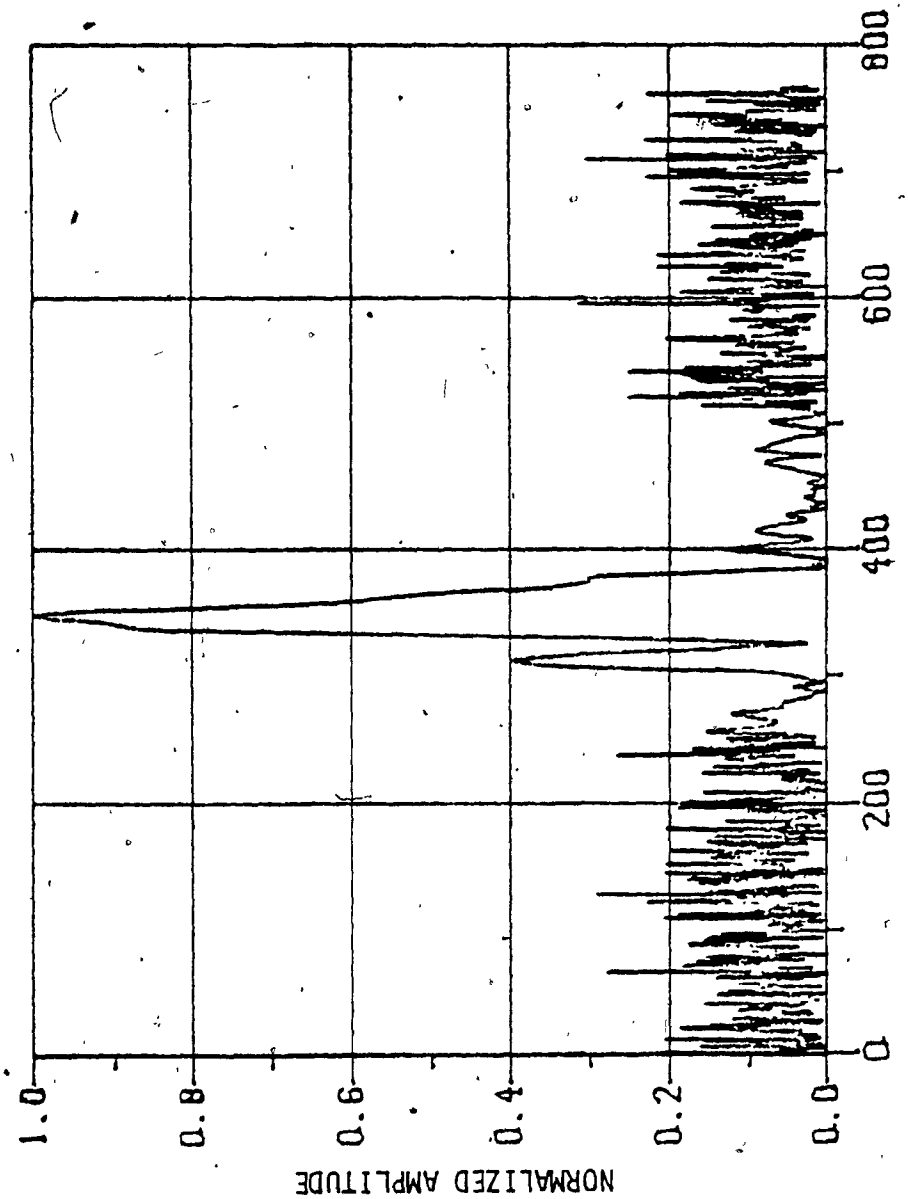


FIGURE 5.6: INPUT BOTTOM RETURN SIGNAL, WITH NOISE FOR ϕ = 15 deg.

APPENDIX B

APPENDIX B

FREQUENCY WINDOWING AND ITS PROPERTIES

In section 3.4.3, the properties of cosine functions are used in frequency windowing with DFT. The windowing in the frequency domain is performed as a spectral convolution. In this Appendix, we will take one cosine window, say Hann window and show how the windowing is performed.

The Hann window in the time domain can be given as:

$$w(t) = 0.5[1-\cos(2\pi t/T_w)] \quad , \quad 0 \leq t \leq T_w \qquad (B1)$$

and for a DFT as

$$w(k) = 0.5[1-\cos(2\pi k/K)] \quad , \quad k=0,1,\ldots,K-1 \qquad (B2)$$

Now transforming equation (B1),

$$W(f) = 0.5[\delta(f)-0.5\{\delta(f-1/T_w)+\delta(f+1/T_w)\}] \qquad (B3)$$

and for DFT

$$W(m) = 0.5[\delta(m)-0.5\{\delta(m-1)+\delta(m+1)\}] \qquad (B4)$$

Then the windowed sequence $H_w(m)$ is obtained as the convolution in the frequency domain as given in equation (3.23), i.e.,

$$H_w(m) = H(m) * W(m) \qquad (B5)$$

By using the sampling property (or shifting property) of the impulse function, we get

$$H_w(m)\Big|_{\text{Hann}} = 0.5[H(m)]-0.25[H(m-1)+H(m+1)]\Big|_{\text{Rectangle}} \qquad (B6)$$

Thus, the windowing is performed by summing the weighted spectral estimates [30]. It must be noted that the rectangle-windowed spectral

points should not be weighted as in equation (B6) when it is interpolated. The attraction of the frequency windows is that the spectra is nonzero at only a few points (three in the case of Hann window) and the windowed spectral points are obtained from the rectangle-windowed spectral points as adds and binary shifts as mentioned in equation (B6).

APPENDIX C

APPENDIX C

USER MANUAL AND LISTING OF COMPUTER PROGRAMS

The computer simulation throughout this investigation is carried out using a Digital computer (CDC CYBER 172). A brief description of the use of programs is outlined and the listings of all the programs are provided in this Appendix. The programs can be used to adopt particular needs by inputting the required data (such as time duration of the transmitted pulse, beamsteering angle and operating depth).

The Fortran programs, subroutines and their functions are listed as follows:

TUKEY	Main Program: Simulates the smooth bottom function and computes the Tukey weighted finite impulse response sequence.
BESEL	Computes the bessel function of the first kind and zero order.
SIG	Main Program: Computes the noise bandlimiting LPF coefficients.
TDRLPF	Generates the bottom return signal with noise.
LAPSAVE	Main Program: Simulates high speed convolution using FFT with overlap-save sectioning.

MODEL — Test Program: Simulates the smooth bottom function.

MPTWN — Test Program: Computes the spectral convolution for frequency windows.

CONVOL — Test Program: Simulates high speed convolution using FFT.

Figure C.1 gives a flow chart of the use of programs.

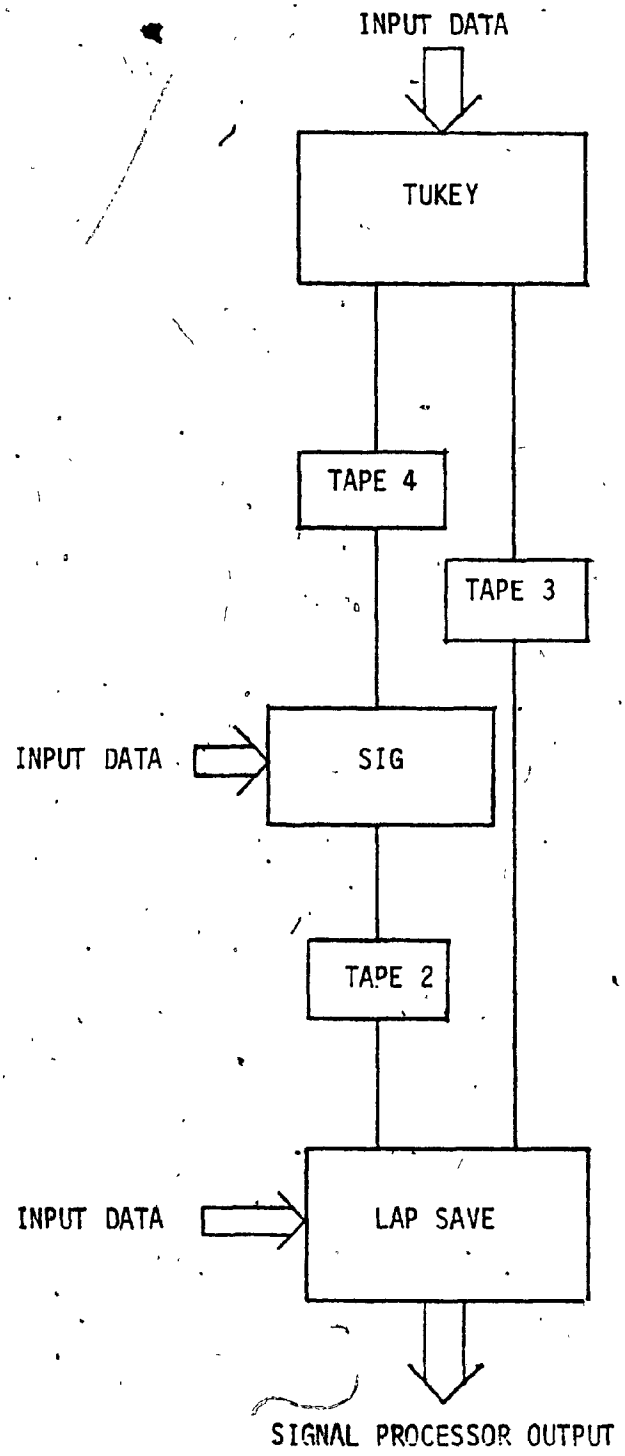


FIGURE C.1: FLOW CHART OF SIMULATION PROGRAM

```
      PROGRAM TUKEY(INPUT,OUTPUT,TAPE3,TAPE4)
C     MAIN PROGRAM-SIMULATION OF SMOOTH BOTTOM FUNCTION
C     AND TUKEY WEIGHTED FIR SEQUENCE.
      DIMENSION IWK(12),AMP(1200),AMAG(1200),PHASE(1200),H(300)
     $,ARAI(1200)
      DIMENSION X1(600),Y1(600,1),Y2(600,1),IMAG4(5151),B(160)
      COMPLEX A(1200)
C     READ INPUT DATA
C     ANG-STEERING ANGLE,(0-45) DEG.
C     TO -TIME DURATION OF THE TRANSMITTED PULSE
C         (10 MS.)
C     D  -DEPTH,12000 FT.
C     N  -NO.OF SAMPLES IN SIGNAL RANGE (TOW).
C     ALPHA1,ALPHA2-TUKEY WINDOW TAPERING COEFF.
      READ*,ANG,TO,D
      READ*,N,TOW
      READ*,ALPHA1,ALPHA2
C     INTERPOLATED BY PADDING ZEROS.
      DO 1 I=1,N
    1 A(I)=(0.0,0.0)
      NN=N/4
      AN1=NN
      AN2=N
      PI=4.*ATAN2(1.0,1.0)
      PHI=PI/180.*ANG
      TT=PI/180.*2.*D/4800.*TAN(PHI)/COS(PHI)+TO
      S=(0.65+ANG/300.)*TT
      EC=TO*(EXP(COS(PHI))-1.)
      DELT=TOW/AN1
      I=1
    4 IF(2.**I.EQ.N)GOTO5
      I=I+1
      GOTO 4
    5 M=I
      DO 10 I=1,12
   10 IWK(I)=0
      T=0.0009
      IF(ANG.GT.0.0) T=0.
      DO 20 I=1,NN
      X=-((T**2)+(EC**2))/2./(S**2)
      Y=(T/S**2)*EXP(X)
      Z=T*EC/S**2
      CALL BESEL(BES,Z)
      H(I)=Y*BES
      T=T+DELT
      HW=H(I)
C     SMOOTH ENVELOPE SEQ.STORED INTAPE4.
      WRITE(4,*) HW
      A(I)=CMPLX(HW,0.)
   20 CONTINUE
```

```
C     COMPUTES FFT
      CALL FFRDR2(A,M,IWK)
      CALL FFT2RV(A,M,IWK)
C     CONJUGATE AND FFT TO GET IFFT.
      DO 25 I=1,N
   25 A(I)=CONJG(A(I))
      CALL FFT2(A,M,IWK)
      CALL FFRDR2(A,M,IWK)
      DO 30 I=1,NN
      AR=REAL(A(I))
      AI=AIMAG(A(I))
      AMP(I)=SQRT(AR**2+AI**2)/AN2
      PRINT*,AMP(I)
   30 CONTINUE
C     THIS PART CONTAINS SIMULATION OF TUKEY WINDOW.
      MM1=INT(ALPHA1*FLOAT(NN))/2
      MM2=INT(ALPHA2*FLOAT(NN))/2
      DO 35 I=1,MM1
      WEIGHT=0.5-0.5*COS(PI*FLOAT(I)/FLOAT(MM1))
      AMP(I)=AMP(I)*WEIGHT
   35 CONTINUE
      DO 36 I=1,MM2
      WEIGHT=0.5-0.5*COS(PI*FLOAT(I)/FLOAT(MM2))
      AMP(NN+1-I)=AMP(NN+1-I)*WEIGHT
   36 CONTINUE
      FREQRES=1./(4.*TOW)
      AMAX=AMP(1)
      DO 99 I=2,NN
      IF(AMAX.LT.AMP(I)) GOTO 98
      GOTO 99
   98 PT=I
      AMAX=AMP(I)
   99 CONTINUE
      DO 37 I=1,N
      A(I)=CMPLX(AMP(I),0.)
      AMP(I)=AMP(I)/AMAX
   37 CONTINUE
      DO 100 I=1,512
C     TUKEY WINDOWED SEQ.STORED IN TAPE3.
      WRITE(3,*) AMP(I)
      PRINT*,I,AMP(I)
  100 CONTINUE
C     THIS PART COMPUTES THE MAG.SPECTRUM OF THE
C     WINDOWED SEQ.
      CALL FFRDR2(A,M,IWK)
      CALL FFT2RV(A,M,IWK)
      DO 38 I=1,N
      ARR=REAL(A(I))
      AII=AIMAG(A(I))
      ARAI(I)=SQRT(ARR*ARR+AII*AII)
      AMAG(I)=20.*ALOG10(ARAI(I)/ARAI(1))
      PHASE(I)=180./PI*ATAN2(AII,ARR)
   38 CONTINUE
```

```
      DO 40 I=1,512
      J=I-1
      FREQ=J*FREQRES
      X1(I)=J
      Y1(I,1)=AMAG(I)
      Y2(I,1)=PHASE(I)
      PRINT 50,J,FREQ,AMAG(I),PHASE(I)
   50 FORMAT(5X,I3,5X,F15.10,5X,E15.8,5X,F10.4)
   40 CONTINUE
      CALL USPLH(X1,Y1,64,1,1,64,B,IMAG4,IER)
      CALL USPLH(X1,Y2,64,1,1,64,B,IMAG4,IER)
      STOP
      END
      SUBROUTINE BESEL(BES,F)
C     COMPUTES BESEL FUNCTION .
      R=F/2.
      T=1.E-10
      E=1.0
      DE=1.0
      DO 5 I=1,25
      DE=DE*R/FLOAT(I)
      SDE=DE**2
      E=E+SDE
      IF(E*T-SDE)5,5,10
    5 CONTINUE
   10 BES=E
      N=I
      RETURN
      END
```

```
      PROGRAM SIGNAL(INPUT,OUTPUT,TAPE4,TAPE2)
C     COMPUTES THE LPF COEFF.
      DIMENSION A0(2),A1(2),A2(2),A3(2),B0(2),B1(2),B2(2),
     $B3(2),CON(2),H(600),HF(600)
      REWIND 4
      DO 5 I=1,512
    5 H(I)=0.0
      PI=4.*ATAN2(1.0,1.0)
      N=256
      DO 1 I=1,N
      READ(4,*) HF(I)
    1 CONTINUE
C     READ PAM - PEAK AMPLITUDE OF THE SMOOTH BOTTOM
C     FUNCTION -READ FROM TAPE4.
      READ*,PAM
      DO 2 I=1,N
      H(I+128)=HF(I)/PAM
    2 CONTINUE
C     TOW- -40DB SIGNAL WIDTH
C     FS - SAMPLING FREQ.
C     FC1 - CUTOFF FREQ. MULTI. NOISE.
C     FC2- CUTOFF FREQ. ADD.NOISE.
      READ*,TOW
      FS=N/TOW
      FC1=50.
      FC2=150.
      WARP1=FS/PI/FC1*TAN(PI*FC1/FS)
      WARP2=FS/PI/FC2*TAN(PI*FC2/FS)
      CON(1)=FS/PI/FC1/WARP1
      CON(2)=FS/PI/FC2/WARP2
      DO 10 J=1,2
      B0(J)=CON(J)**3+2.*CON(J)*CON(J)+2.*CON(J)+1.
      A0(J)=1./B0(J)
      A1(J)=3./B0(J)
      A2(J)=3./B0(J)
      A3(J)=1./B0(J)
      B1(J)=(-3.*CON(J)**3-2.*CON(J)*CON(J)+2.*CON(J)+3.)/B0(J)
      B2(J)=(3.*CON(J)**3-2.*CON(J)*CON(J)-2.*CON(J)+3.)/B0(J)
      B3(J)=(-CON(J)**3+2.*CON(J)*CON(J)-2.*CON(J)+1.)/B0(J)
   10 CONTINUE
      DO 20 J=1,2
      PRINT*,A0(J),A1(J),A2(J),A3(J)
      PRINT*,B1(J),B2(J),B3(J)
   20 CONTINUE
      CALL TDRLPF(A0,A1,A2,A3,B1,B2,B3,H,N,FS)
      ?
```

```
      STOP
      END
      SUBROUTINE TDRLPF(A0,A1,A2,A3,B1,B2,B3,H,N,FS)
C     GENERATES THE BOTTOM RETURN SIGNAL WITH NOISE.
      DIMENSION A0(2),A1(2),A2(2),A3(2),B1(2),B2(2),B3(2),HF(1200)
     $,R(600),X(600),Y(600),H(600),S(600),T(600,1),IMAG4(5151)
      DO 5 I=1,3
      X(I)=0.0
      Y(I)=0.0
    5 CONTINUE
      ISEED=1
      M=512
      MM=768
C     GGNOR-RANDOM NOISE GEN.-WHITE SEQ. WITH MEAN ZERO
C     AND VARIANCE SIGMA**2.
      CALL GGNOR(ISEED,M,R)
      CALL FTAUTO(R,M,0,0,1,AMEAN,VAR,0,0,0,0)
      CALL GGNOR(ISEED,MM,HF)
      CALL FTAUTO(HF,MM,0,0,1,AMOM,DAD,0,0,0,0)
      DO 12 I=1,MM
      HF(I)=ABS(HF(I)-AMOM)/SQRT(DAD*100.)
   12 CONTINUE
      DO 30 J=1,2
      DO 30 I=1,M
      K=I+3
      IF(J.EQ.1) GOTO 2
      IF(J.EQ.2) GOTO 3
    2 X(K)=(R(I)-AMEAN)/SQRT(VAR)
      GOTO 10
    3 X(K)=(R(I)-AMEAN)/SQRT(VAR*100.)
      GOTO 20
   10 Y(K)=A0(J)*X(K)+A1(J)*X(K-1)+A2(J)*X(K-2)+A3(J)*X(K-3)-
     $B1(J)*Y(K-1)-B2(J)*Y(K-2)-B3(J)*Y(K-3)
      H(I)=H(I)*Y(K)
      GOTO 30
   20 Y(K)=A0(J)*X(K)+A1(J)*X(K-1)+A2(J)*X(K-2)+A3(J)*X(K-3)-
     $B1(J)*Y(K-1)-B2(J)*Y(K-2)-B3(J)*Y(K-3)
      H(I)=ABS(H(I)+Y(K))
   30 CONTINUE
      HMAX=H(1)
      DO 65 I=2,M
      IF(HMAX.LT.H(I)) GOTO 60
      GOTO 65
   60 PT=I
      HMAX=H(I)
```

```
      65  CONTINUE
C     NORMALIZING
      DO 70 I=128,384
      HF(I+128)=H(I)/HMAX
   70 CONTINUE
      DO 100 I=1,MM
      S(I)=I/FS
      T(I,1)=HF(I)
C      BOTTOM RETURN SIGNAL WITH NOISE-STORED IN TAPE2
      WRITE(2,*) HF(I)
      PRINT*,I,HF(I)
  100 CONTINUE
      CALL USPLH(S,T,512,1,1,512,A,IMAG4,IER)
      STOP
      END
?
```

```
      PROGRAM LAPSAVE(INPUT,OUTPUT,TAPE2,TAPE3)
C     SIMULATION OF HIGH SPEED CONVOLUTION USING FFT
C     AND 'SECTIONING'- OVERLAP-SAVE METHOD.
      DIMENSION IWK(10),AMP(600),H(900),HW(600),W(1200)
     $,SEC1(800),SEC2(800),SEC3(800),OUT(800)
      COMPLEX A(600),V(600),Y(600),ZZ
C     READ INPUT DATA
C     N-TRANSFORM LENGTH:TOW-SIGNAL LENGTH IN S.
      READ*,N,TOW
      DO 1 I=1,N
      V(I)=(0.0,0.0)
      A(I)=(0.0,0.0)
      W(I)=0.0
    1 CONTINUE
      REWIND 2
      REWIND 3
      DO 2 I=1,N
      READ(3,*) AMP(I)
    2 CONTINUE
      DO 3 I=1,768
      READ(2,*) H(I)
      W(I+256)=H(I)
    3 CONTINUE
      AN=N
      FREQRES=1./(2.*TOW)
      TIME=TOW/256.
      PI=4.*ATAN2(1.0,1.0)
      I=1
    4 IF(2.**I.EQ.N)GOTO5
      I=I+1
      GOTO 4
    5 M=I
      DO 10 I=1,10
   10 IWK(I)=0
      DO 99 J=1,3
      DO 20 I=1,N
      II=I+(J-1)*256
      HW(I)=W(II)
      A(I)=CMPLX(HW(I),0.)
   20 CONTINUE
      DO 22 I=1,N
   22 PRINT*,I,HW(I)
      CALL FFRDR2(A,M,IWK)
      CALL FFT2RV(A,M,IWK)
      DO 30 I=1,N
```

```
          V(I)=CMPLX(AMP(I),0.)
   30   CONTINUE
        CALL FFRDR2(V,M,IWK)
        CALL FFT2RV(V,M,IWK)
        TS=TOW
        W=2.*PI*FREQRES
        DO 40 I=1,N
        L=(I-1)*(-1)
        ZZ=CMPLX(COS(L*W*TS),SIN(L*W*TS))
        V(I)=V(I)*ZZ
   40   CONTINUE
        DO 45 I=1,N
        V(I)=CONJG(V(I))
        Y(I)=A(I)*V(I)
        Y(I)=CONJG(Y(I))
   45   CONTINUE
        CALL FFT2(Y,M,IWK)
        CALL FFRDR2(Y,M,IWK)
        DO 80 I=1,N
        AR=REAL(Y(I))
        AI=AIMAG(Y(I))
        IF(J.EQ.2) GOTO 50
        IF(J.EQ.3) GOTO 60
        SEC1(I)=SQRT(AR*AR+AI*AI)/AN
        GOTO 80
   50   SEC2(I)=SQRT(AR*AR+AI*AI)/AN
        GOTO 80
   60   SEC3(I)=SQRT(AR*AR+AI*AI)/AN
   80   CONTINUE
   99   CONTINUE
        DO 100 I=1,768
        IF(I.LT.256)OUT(I)=SEC1(I+256)
        IF((I.GE.256).AND.(I.LT.512))OUT(I)=SEC2(I)
        IF(I.GE.512)OUT(I)=SEC3(I-256)
  100   CONTINUE
        DO 120 I=1,768
        STIME=I*TIME
        PRINT*,I,STIME,OUT(I)
  120   CONTINUE
        STOP
        END
*41 LINES.
*?
```

```
      PROGRAM MODEL(INPUT,OUTPUT,TAPE1)
C     THIS PROGRAM SIMULATES THE SMOOTH BOTTOM FUNCTION.
      DIMENSION X1(401),Y1(401,1),IMAG4(5151),A(160)
C     READ INPUT DATA
C     ANG-BEAM STEERING ANGLE- (0-45)DEG.
C     TO -TIME DURATION OF THE TRANSMITTED PULSE-10 MS.
C     D  -OPERATING DEPTH-12000 FT.
      READ*,ANG
      READ*,TO
      READ*,D
      PI=4.*ATAN2(1.0,1.0)
      PHI=PI/180.*ANG
      TT=PI/180.*2.*D/4800.*TAN(PHI)/COS(PHI)+TO
      S=(0.65+ANG/300.)*TT
      EC=TO*(EXP(COS(PHI))-1.)
      DO 10 I=1,401
      T=(I-1.)*0.0001
      X1(I)=T
      X=-((T**2)+(EC**2))/2./(S**2)
      Y=(T/S**2)*EXP(X)
      Z=T*EC/S**2
      CALL BESEL(BES,Z)
      HT=Y*BES
      Y1(I,1)=HT
      WRITE (1,*)T,HT
   10 CONTINUE
      STOP
      END
      SUBROUTINE BESEL(BES,F)
C     CALCULATES THE BESSEL FUNCTION OF THE FIRST
C     KIND AND ZERO ORDER.
      R=F/2.
      T=1.E-10
      E=1.0
      DE=1.0
      DO 5 I=1,25
      DE=DE*R/FLOAT(I)
      SDE=DE**2
      E=E+SDE
      IF(E*T-SDE)5,5,10
    5 CONTINUE
   10 BES=E
      N=I
      RETURN
      END
```

```
      PROGRAM MFTWW(INPUT,OUTPUT)
C     COMPUTES SPECTRAL CONVOLUTION OF THE
C     FREQ.WINDOWS.
      DIMENSION IWK(12),AMAG(600),PHASE(600),H(300),AM(600)
      DIMENSION X1(600),Y1(600,1),Y2(600,1),IMAG4(5151),B(160)
      COMPLEX A(600),ZZ,V(300),W(1200)
C     READ INPUT DATA
      READ*,ANG,TO,D
      READ*,N,TOW
      READ*,W1,W2,W3,W4,W5
      DO 1 I=1,1024
    1 W(I)=(0.,0.)
      K=N/2
      MM=10
      AN=N
      PI=4.*ATAN2(1.0,1.0)
      PHI=PI/180.*ANG
      TT=PI/180.*2.*D/4800.*TAN(PHI)/COS(PHI)+TO
      S=(0.65+ANG/300.)*TT
      EC=TO*(EXP(COS(PHI))-1.)
      DELT=TOW/AN
      I=1
    4 IF(2.**I.EQ.N)GOTO5
      I=I+1
      GOTO 4
    5 M=I
      DO 10 I=1,12
   10 IWK(I)=0
      T=0.
      IF(ANG.EQ.0.) T=0.0009
      DO 20 I=1,N
      X=-((T**2)+(EC**2))/2./(S**2)
      Y=(T/S**2)*EXP(X)
      Z=T*EC/S**2
      CALL BESEL(BES,Z)
      H(I)=Y*BES
      T=T+DELT
      HW=H(I)
      A(I)=CMPLX(HW,0.)
   20 CONTINUE
      CALL FFRDR2(A,M,IWK)
      CALL FFT2RV(A,M,IWK)
      DO 25 I=1,N
      IF(I.EQ.1) GOTO 22
      IF(I.EQ.N) GOTO 23
      V(I)=A(I-1)*W1+A(I)*W2+A(I+1)*W3
      GOTO 25
   22 V(I)=A(N)*W1+A(I)*W2+A(I+1)*W3
      GOTO 25
   23 V(I)=A(I-1)*W1+A(I)*W2+A(1)*W3
> ?
```

```
      25  CONTINUE
          DO 30 I=1,N
          IF(I.EQ.N) GOTO 28
          V(I)=V(I)*W4+V(I+1)*W5
          GOTO 30
      28  V(I)=V(I)*W4+V(1)*W5
      30  CONTINUE
          DO 32 I=1,N
      32  V(I)=CONJG(V(I))
          CALL FFT2(V,M,IWK)
          CALL FFRDR2(V,M,IWK)
          DO 35 I=1,N
          AMP=CABS(V(I))/AN
          PRINT*,I,AMP
          W(I)=CMPLX(AMP,0.0)
      35  CONTINUE
          CALL FFRDR2(W,MM,IWK)
          CALL FFT2RV(W,MM,IWK)
          DO 38 I=1,512
          AR=REAL(W(I))
          AI=AIMAG(W(I))
          AM(I)=SQRT(AR*AR+AI*AI)
          AMAG(I)=20.*ALOG10(AM(I)/AM(1))
          PHASE(I)=180./PI*ATAN2(AI,AR)
      38  CONTINUE
          FREQRES=1./(4.*TOW)
          DO 40 I=1,512
          J=I-1
          FREQ=J*FREQRES
          X1(I)=J
          Y1(I,1)=AMAG(I)
          Y2(I,1)=PHASE(I)
          PRINT 50,J,FREQ,AMAG(I),PHASE(I)
      50  FORMAT(5X,I3,5X,F15.10,5X,E15.8,5X,F10.4)
      40  CONTINUE
          CALL USPLH(X1,Y1,64,1,1,64,B,IMAG4,IER)
          CALL USPLH(X1,Y2,64,1,1,64,B,IMAG4,IER)
          STOP
          END
          SUBROUTINE BESEL(BES,F)
          R=F/2.
          T=1.E-10
          E=1.0
          DE=1.0
          DO 5 I=1,25
          DE=DE*R/FLOAT(I)
          SDE=DE**2
          E=E+SDE
          IF(E*T-SDE)5,5,10
       5  CONTINUE
      10  BES=E
          N=I
          RETURN
          END
     > ?
```

```
      PROGRAM CONVOL(INPUT,OUTPUT,TAPE2,TAPE3)
C     TEST PROGRAM TO SIMULATE THE HIGH SPEED CONVOLUTION.
      DIMENSION IWK(10),AMP(600),H(600),X1(600),
     $Y1(600,1),IMAG4(5151)
      COMPLEX A(600),V(600),Y(600),ZZ
      READ*,N,TOW
      DO 1 I=1,N
      H(I)=(0.0,0.0)
      V(I)=(0.0,0.0)
    1 CONTINUE
      REWIND 2
      REWIND 3
      DO 2 I=1,N
      READ(2,*) H(I)
      READ(3,*) AMP(I)
    2 CONTINUE
      AN=N
      FREQRES=1./(4.*TOW)
      TIME=TOW/256.
      PI=4.*ATAN2(1.0,1.0)
      I=1
    4 IF(2.**I.EQ.N)GOTO5
      I=I+1
      GOTO 4
    5 M=I
      DO 10 I=1,10
   10 IWK(I)=0
      DO 20 I=1,N
      HW=H(I)
      PRINT*,I,HW
      A(I)=CMPLX(HW,0.)
   20 CONTINUE
      CALL FFRDR2(A,M,IWK)
      CALL FFT2RV(A,M,IWK)
      DO 30 I=1,N
      V(I)=CMPLX(AMP(I),0.)
   30 CONTINUE
      CALL FFRDR2(V,M,IWK)
      CALL FFT2RV(V,M,IWK)
      TS=TOW
      W=2.*PI*FREQRES
      DO 40 I=1,N
      L=(I-1)*(-1)
      ZZ=CMPLX(COS(L*W*TS),SIN(L*W*TS))
      V(I)=V(I)*ZZ
> ?
```

```
   40  CONTINUE
       DO 45 I=1,N
       V(I)=CONJG(V(I))
       Y(I)=A(I)*V(I)
       Y(I)=CONJG(Y(I))
   45  CONTINUE
       CALL FFT2(Y,M,IWK)
       CALL FFRDR2(Y,M,IWK)
       DO 50 I=1,N
       AR=REAL(Y(I))
       AI=AIMAG(Y(I))
       OUT=SQRT(AR*AR+AI*AI)/AN
       STIME=I*TIME
       X1(I)=STIME
       Y1(I,1)=OUT
       PRINT*,I,STIME,OUT
   50  CONTINUE
       CALL USPLH(X1,Y1,512,1,1,512,A,IMAG4,IER)
       STOP
       END
> ?
```